



TUGAS AKHIR - KI141502

RANCANG BANGUN PERANGKAT LUNAK SISTEM INFORMASI AKADEMIK GENERIK BERBASIS WEB PADA MODUL EKUIVALENSI

Bustan Amal Alfirdaus
NRP 5111 100 019

Dosen Pembimbing
Umi Laili Yuhana, S.Kom., M.Sc.
Ratih Nur Esti Anggraini, S.Kom., M.Sc.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2015



UNDERGRADUATE THESES - KI141502

SOFTWARE DESIGN AND IMPLEMENTATION OF WEB BASED GENERIC ACADEMIC INFORMATION SYSTEM FOR EQUIVALENCE MODULE

Bustan Amal Alfirdaus
NRP 5111 100 019

Advisor
Umi Laili Yuhana, S.Kom., M.Sc.
Ratih Nur Esti Anggraini, S.Kom., M.Sc.

INFORMATICS ENGINEERING DEPARTEMENT
Information Technology Faculty
Sepuluh Nopember Institute of Technology
Surabaya 2015

LEMBAR PENGESAHAN

RANCANG BANGUN PERANGKAT LUNAK SISTEM INFORMASI AKADEMIK GENERIK BERBASIS WEB PADA MODUL EKVIVALENSI

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Rekayasa Perangkat Lunak
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :
BUSTAN AMAL ALFIRDAUS
NRP : 5111 100 019

Disetujui oleh Dosen Pembimbing Tugas Akhir :

Umi Laili Yuhana, S.Kom., M.Sc.
NIP: 197906 26200501 2 002 (pembimbing 1)

Ratih Nur Esti Anggraini, S.Kom., M.Sc.
NIP: 198412 10201404 2 003 (pembimbing 2)

SURABAYA
JUNI 2015

RANCANG BANGUN PERANGKAT LUNAK SISTEM INFORMASI AKADEMIK GENERIK BERBASIS WEB PADA MODUL EQUIVALENSI

Nama Mahasiswa : Bustan Amal Alfirdaus
NRP : 5111 100 019
Jurusan : Teknik Informatika FTIf-ITS
Dosen Pembimbing : Umi Laili Yuhana, S.Kom., M.Sc.
Ratih Nur Esti Anggraini, S.Kom., M.Sc.

ABSTRAK

Indonesia memiliki berbagai macam perguruan tinggi yang dibedakan berdasarkan jenjang dan program studi yang ditawarkan seperti universitas, institusi, sekolah tinggi, politeknik, akademi, dan akademi komunitas. Dari keberagaman tersebut tentunya tiap-tiap perguruan tinggi memiliki proses kegiatan akademik yang berbeda, terutama dalam proses ekuivalensi. Ekuivalensi merupakan proses penyetaraan matakuliah. Kasus yang sering terjadi yaitu saat pergantian kurikulum dan penerimaan peserta didik alih jenjang. Pada kasus pergantian kurikulum, tidak semua perguruan tinggi memiliki sistem yang dapat menangani hal tersebut. Sedangkan kasus penerimaan peserta didik alih jenjang masih dilakukan secara manual oleh kepala program studi. Sehingga diperlukan sebuah sistem yang dapat mengakomodasi kebutuhan perguruan tinggi.

Melalui Tugas Akhir ini dibangun sebuah sistem informasi akademik generik pada modul ekuivalensi. Generik yang dimaksud yaitu memiliki proses bisnis yang generik atau umum. Proses bisnis yang hanya dimiliki oleh perguruan tinggi tertentu akan diabaikan jika mengganggu proses bisnis yang lain yang dianggap lebih generik.

Pengujian dilakukan dengan bantuan dua dosen Teknik Informatika yang tergabung dalam Tim Ekuivalensi. Hasilnya sistem ini sesuai dengan proses ekuivalensi yang ada pada ITS. Dengan begitu diharapkan sistem ini dapat menangani variasi aturan yang dimiliki oleh perguruan tinggi yang lain.

Kata kunci : Ekuivalensi, Generik, Sistem Informasi Akademik

SOFTWARE DESIGN AND IMPLEMENTATION OF WEB BASED GENERIC ACADEMIC INFORMATION SYSTEM FOR EQUIVALENCE MODULE

Student Name : Bustan Amal Alfirdaus
NRP : 5111 100 019
Major : Teknik Informatika FTIf-ITS
Advisor : Umi Laili Yuhana, S.Kom., M.Sc.
Ratih Nur Esti Anggraini, S.Kom., M.Sc.

ABSTRACT

Indonesia has a wide range of universities, differentiated by level and concentration offered such as university, institution, college, polytechnic, academy, and community academy. From the diversity, each university has a different process of academic activities, especially in process of equivalence. Equivalence is a process of equalizing the courses change. The problem mostly occurs when curriculum and admission of transfer students happened. In the case of alternation of curriculum, not all universities have a system that can handle it. While in the case of admission of transfer students, it is still done manually by head of departement. Hence, we need a system that can accommodate this needs of the university.

In this final project, we build a generic academic information system that focused on equivalence module. Generic is referred to a generic business process or general business process. The business process only owned by specific college will be ignored if it ruins another business process which is considered to be important.

The trial will be done with the help of two of Department of Informatics lecturers. Both are the members of Equivalence Team. The result shows that the system is in line with the process

of equivalence at ITS. So, we are hoped that the system can handle variation of rules that are owned by other university.

Keyword: Equivalence, Generic, Academic Information System

KATA PENGANTAR

Alhamdulillahirabbil'alamin, segala puji bagi Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul **“RANCANG BANGUN PERANGKAT LUNAK SISTEM INFORMASI AKADEMIK GENERIK BERBASIS WEB PADA MODUL EKUIVALENSI”**.

Tugas Akhir ini merupakan salah satu syarat untuk memperoleh gelar Sarjana Komputer di Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Teknologi Sepuluh Nopember Surabaya. Dengan Tugas Akhir ini penulis juga dapat menghasilkan suatu implementasi dari apa yang telah penulis pelajari.

Selesainya Tugas Akhir ini tidak lepas dari bantuan dan dukungan beberapa pihak. Sehingga pada kesempatan ini penulis mengucapkan syukur dan terima kasih kepada:

1. Allah SWT dan Nabi Muhammad SAW.
2. Kedua orang tua, dan keluarga penulis, terima kasih atas doa dan bantuan moral dan material selama penulis belajar di Teknik Informatika ITS.
3. Direktorat Jenderal Pendidikan Tinggi, terima kasih atas beasiswa bidik misi yang diberikan selama penulis belajar di Teknik Informatika ITS.
4. Ibu Umi Laili Yuhana, S.Kom., M.Sc. selaku pembimbing I dan membantu sekaligus membimbing penulis selama pengerjaan Tugas Akhir.
5. Ibu Ratih Nur Esti Anggrain, S.Kom., M.Sc. selaku pembimbing II dan membantu sekaligus membimbing penulis selama pengerjaan Tugas Akhir.
6. Ibu Dr. Eng. Nanik Suciati, S.Kom., M.Kom. selaku Kepala Jurusan Teknik Informatika ITS, dan Bapak Radityo Anggoro, S.Kom., M.Sc. selaku koordinator TA.

7. Seluruh narasumber dari UNESA, UNAIR, ITS, UNNAR, PENS dan UPN yang telah meluangkan waktu saat penggalian proses bisnis, serta peserta uji coba (Ibu Dini dan Bapak Radityo Anggoro) yang sudah meluangkan waktu juga di tengah-tengah kesibukan Bapak Ibu.
8. Bapak dan Ibu Dosen di Jurusan Teknik Informatika yang telah memberikan ilmu selama penulis belajar di Teknik Informatika ITS.
9. Seluruh staf dan karyawan Teknik Informatika yang telah memberikan bantuan selama penulis belajar di Teknik Informatika ITS.
10. Rekan “TC Hura-Hura” yang sudah menemani penulis cangkruk, mbolang, dan hura-hura selama di Teknik Informatika.
11. Kelompok TA “bit.ly/driveTA” Galih, Tommy, Amanda, Rahman, dan Agung, yang sudah berbagi ilmu selama pengerjaan Tugas Akhir ini.
12. Rekan “TC 2011” yang telah membantu, berbagi ilmu, menjaga kebersamaan, dan memberi motivasi kepada penulis. Rekan di Laboratorium ALPRO dan RPL yang menemani penulis di lab selama pengerjaan Tugas Akhir.
13. Serta semua pihak yang telah turut membantu penulis secara langsung maupun tidak langsung dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Sehingga dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depannya.

Surabaya, Juni 2015

Bustan Amal Alfirdaus

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR.....	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
DAFTAR KODE SUMBER	xxiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan	4
1.5 Manfaat	4
1.6 Metodologi	4
1.7 Sistematika Penulisan	6
BAB II TINJAUAN PUSTAKA	9
2.1 Penelitian Terkait	9
2.2 Perangkat Lunak Generik.....	10
2.3 Ekuivalensi.....	10
2.4 Kerangka Kerja Spring.....	11
2.4.1 JDBC (<i>Java Database Connectivity</i>) [11]	12
2.4.2 <i>Spring-Transaction Manager</i> [10]	13
2.4.3 Servlet [13]	14
2.4.4 Beans [11]	15
2.5 PostgreSQL [12]	15
BAB III ANALISIS DAN PERANCANGAN	17
3.1 Analisis.....	17
3.1.1 Penggalan Proses Bisnis Modul Ekuivalensi	17
3.1.2 Deskripsi Umum Sistem.....	23
3.1.3 Spesifikasi Kebutuhan Perangkat Lunak.....	23
3.1.3.1 Kebutuhan Fungsional.....	24

3.1.3.2	Aktor	25
3.1.3.3	Kasus Penggunaan	25
3.2	Perancangan Sistem	32
3.2.1	Perancangan Basis Data	33
3.2.2	Perancangan Antarmuka	35
3.2.2.1	Tabel Master	35
3.2.2.2	<i>Form Detail</i>	36
BAB IV	IMPLEMENTASI	39
4.1	Lingkungan Pembangunan Perangkat Lunak	39
4.2	Implementasi Kelas	39
4.2.1	Implementasi Kelas Entitas	40
4.2.2	Implementasi Kelas <i>Repository</i>	42
4.2.3	Implementasi Kelas <i>Service</i>	43
4.2.4	Implementasi Kelas <i>Controller</i>	45
4.3	Implementasi Proses Bisnis	45
4.3.1	Mengelola Matakuliah Alih Jenjang	46
4.3.2	Pemetaan Matakuliah	46
4.3.3	Pemrosesan aturan ekuivalensi	46
4.4	Implementasi Antarmuka Pengguna	46
4.4.1	Implementasi Tabel Master	47
4.4.2	Halaman Pemetaan Matakuliah	47
4.4.3	Implementasi <i>Form Detail</i>	48
4.4.4	Halaman Proses Ekuivalensi Alih Jenjang	48
4.4.5	Halaman Kelola Matakuliah Wajib	48
4.4.6	Halaman Proses Ekuivalensi Kurikulum	49
BAB V	PENGUJIAN DAN EVALUASI	51
5.1	Lingkungan Uji Coba	51
5.2	Skenario Pengujian	51
5.2.1	Pengujian Fungsionalitas	52
5.2.1.1	Pengujian fungsionalitas mengelola katalog alih jenjang	54
5.2.1.2	Pengujian fungsionalitas mengelola matakuliah alih jenjang	54
5.2.1.3	Pengujian fungsionalitas membagikan katalog ke satuan manajemen	55

5.2.1.4	Pengujian fungsionalitas mengelola peserta didik alih jenjang.....	56
5.2.1.5	Pengujian fungsionalitas ekuivalensi peserta didik alih jenjang.....	57
5.2.1.6	Pengujian fungsionalitas verifikasi matakuliah wajib.....	58
5.2.1.7	Pengujian fungsionalitas mencetak laporan ekuivalensi alih jenjang.....	58
5.2.1.8	Pengujian fungsionalitas mencetak laporan ekuvialensi	59
5.2.1.9	Pengujian fungsionalitas verifikasi ekuivalensi	60
5.2.1.10	Pengujian fungsionalitas mengelola aturan ekuivalensi kurikulum.....	60
5.2.1.11	Pengujian fungsionalitas mengelola aturan ekuivalensi alih jenjang.....	61
5.2.2	Perbedaan Aturan Ekuivalensi	62
5.2.3	Proses Pemadanan Matakuliah.....	62
BAB VI KESIMPULAN DAN SARAN.....		65
6.1	Kesimpulan	65
6.2	Saran	65
DAFTAR PUSTAKA.....		67
LAMPIRAN A. KODE SUMBER.....		69
LAMPIRAN B. TABEL.....		89
LAMPIRAN C. GAMBAR.....		99
BIODATA PENULIS.....		121

DAFTAR TABEL

Tabel 3.1 Daftar sampel perguruan tinggi	18
Tabel 3.2 Hasil wawancara proses bisnis	19
Tabel 3.3 Hasil wawancara proses bisnis (2)	20
Tabel 3.4 Kebutuhan fungsional	24
Tabel 3.5 Kebutuhan fungsional (2).....	25
Tabel 3.6 Kasus penggunaan mengelola aturan ekuivalensi alih jenjang	27
Tabel 3.7 Kasus penggunaan mengelola aturan ekuivalensi alih jenjang (2)	28
Tabel 3.8 Contoh aturan ekuivalensi	29
Tabel 3.9 Contoh aturan ekuivalensi	29
Tabel 3.10 Contoh aturan ekuivalensi	29
Tabel 4.1 Fungsi umum kelas <i>repository</i>	43
Tabel 4.2 Fungsi umum kelas <i>service</i>	44
Tabel 4.3 <i>Property</i> umum kelas <i>service</i>	45
Tabel 5.1 Kebutuhan fungsional untuk tim ekuivalensi	52
Tabel 5.2 Kebutuhan fungsional untuk peserta didik.....	52
Tabel 5.3 Kebutuhan fungsional untuk kepala satuan manajemen	53
Tabel 5.4 Rincian pengujian fungsionalitas mengelola katalog alih jenjang	54
Tabel 5.5 Rincian pengujian fungsionalitas mengelola matakuliah alih jenjang	54
Tabel 5.6 Rincian pengujian fungsionalitas mengelola matakuliah alih jenjang (2)	55
Tabel 5.7 Rincian pengujian fungsionalitas membagikan katalog ke satuan manajemen.....	55
Tabel 5.8 Rincian pengujian fungsionalitas membagikan katalog ke satuan manajemen (2).....	56
Tabel 5.9 Rincian pengujian fungsionalitas mengelola peserta didik alih jenjang	56

Tabel 5.10 Rincian pengujian fungsionalitas mengelola peserta didik alih jenjang (2).....	57
Tabel 5.11 Rincian pengujian fungsionalitas ekuivalensi peserta didik alih jenjang.....	57
Tabel 5.12 Rincian pengujian fungsionalitas verifikasi matakuliah wajib.....	58
Tabel 5.13 Rincian pengujian fungsionalitas mencetak laporan ekuivalensi alih jenjang.....	58
Tabel 5.14 Rincian pengujian fungsionalitas mencetak laporan ekuivalensi alih jenjang (2).....	59
Tabel 5.15 Rincian pengujian fungsionalitas mencetak laporan ekuivalensi	59
Tabel 5.16 Rincian pengujian fungsionalitas verifikasi ekuivalensi	60
Tabel 5.17 Rincian pengujian fungsionalitas mengelola aturan ekuivalensi kurikulum.....	60
Tabel 5.18 Rincian pengujian fungsionalitas mengelola aturan ekuivalensi kurikulum (2).....	61
Tabel 5.19 Rincian pengujian fungsionalitas mengelola aturan ekuivalensi alih jenjang.....	61
Tabel 5.20 Skenario matakuliah memiliki prasyarat.....	62
Tabel 5.21 Skenario matakuliah memiliki prasyarat (2).....	63
Tabel 5.22 Skenario pemadanan matakuliah	63
Tabel B.1 Rincian kasus penggunaan kelola katalog alih jenjang	89
Tabel B.2 Rincian kasus penggunaan kelola matakuliah alih jenjang.....	90
Tabel B.3 Rincian kasus penggunaan membagikan katalog ke satuan manajemen	91
Tabel B.4 Rincian kasus penggunaan mengelola peserta didik alih jenjang.....	92
Tabel B.5 Rincian kasus penggunaan mengelola ekuivalensi peserta didik alih jenjang	93
Tabel B.6 Rincian kasus penggunaan verifikasi matakuliah wajib	94

Tabel B.7 Rincian kasus penggunaan mencetak laporan Ekuivalensi alih jenjang	95
Tabel B.8 Rincian kasus penggunaan mengelola aturan ekuivalensi	96
Tabel B.9 Rincian kasus penggunaan verifikasi ekuivalensi	97
Tabel B.10 Rincian kasus penggunaan mencetak laporan ekuivalensi.....	97

DAFTAR GAMBAR

Gambar 1.1 Jumlah perguruan tinggi di Indonesia [3].....	1
Gambar 2.1 Gambaran umum kerangka kerja spring [8]	11
Gambar 2.2 Arsitektur JDBC pada arsitektur <i>Two-Tier</i> [11]	12
Gambar 2.3 Arsitektur JDBC pada arsitektur <i>Three-Tier</i> [11]...	13
Gambar 3.1 Rangkaian tahapan analisis proses bisnis	17
Gambar 3.2 Alur proses bisnis ekuivalensi alih jenjang	22
Gambar 3.3 Alur proses bisnis ekuivalensi kurikulum	22
Gambar 3.4 Kasus penggunaan modul ekuivalensi.....	26
Gambar 3.5 Kasus penggunaan mengelola aturan ekuivalensi alih jenjang	30
Gambar 3.6 <i>Pseudocode</i> metode <i>infix to postfix</i>	30
Gambar 3.7 <i>Pseudocode</i> metode <i>infix to postfix</i> (2).....	31
Gambar 3.8 Desain tabel master.....	36
Gambar 3.9 Desain <i>form detail</i>	37
Gambar 3.10 Desain halaman pemetaan matakuliah	37
Gambar 3.11 Desain halaman kelola matakuliah wajib	38
Gambar 4.1 Implementasi antarmuka tabel master	47
Gambar 4.2 Implementasi antarmuka halaman pemetaan matakuliah	47
Gambar 4.3 Implementasi antarmuka <i>form detail</i>	48
Gambar 4.4 Implementasi antarmuka halaman ekuivalensi alih jenjang	49
Gambar 4.5 Implementasi antarmuka halaman kelola matakuliah wajib.....	49
Gambar 4.6 Implementasi antarmuka halaman proses ekuivalensi kurikulum	50
Gambar 5.1 Hasil skenario matakuliah memiliki prasyarat	64
Gambar 5.2 Hasil skenario pemadanan matakuliah	64
Gambar C.1 Kasus penggunaan mengelola katalog alih jenjang	99
Gambar C.2 Kasus penggunaan mengelola matakuliah alih jenjang	100

Gambar C.3 Kasus penggunaan membagikan katalog ke satuan manajemen.....	101
Gambar C.4 Kasus penggunaan mengelola peserta didik alih jenjang.....	102
Gambar C.5 Kasus penggunaan mengelola ekuivalensi peserta didik alih jenjang.....	103
Gambar C.6 Kasus penggunaan verifikasi matakuliah wajib ...	104
Gambar C.7 Kasus penggunaan mencetak laporan ekuivalensi alih jenjang.....	105
Gambar C.8 Kasus penggunaan mengelola aturan ekuivalensi	106
Gambar C.9 Kasus penggunaan verifikasi ekuivalensi.....	107
Gambar C.10 Kasus penggunaan mencetak laporan ekuivalensi	108
Gambar C.11 Lembar <i>feedback</i>	109
Gambar C.12 Lembar <i>feedback</i>	110
Gambar C.13 Lembar kuisioner.....	111
Gambar C.14 Lembar kuisioner.....	112
Gambar C.15 Lembar kuisioner.....	113
Gambar C.16 Lembar kuisioner.....	114
Gambar C.17 Lembar kuisioner.....	115
Gambar C.18 Lembar kuisioner.....	116
Gambar C.19 CDM ekuivalensi alih jenjang	117
Gambar C.20 PDM ekuivalensi alih jenjang	118
Gambar C.21 CDM ekuivalensi kurikulum	119
Gambar C.22 PDM ekuivalensi kurikulum.....	120

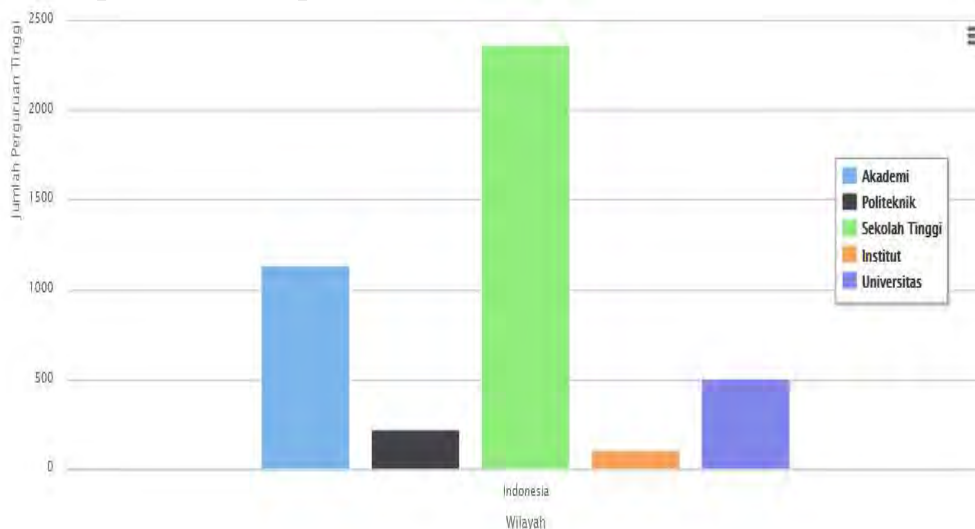
BAB I

PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar tugas akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan tugas akhir, dan sistematika penulisan.

1.1 Latar Belakang

Pendidikan tinggi adalah pendidikan pada jalur pendidikan sekolah pada jenjang yang lebih tinggi daripada pendidikan menengah di jalur pendidikan sekolah [1]. Institusi pendidikan tinggi yang menawarkan pendidikan akademik dan vokasi dapat dibedakan berdasarkan jenjang dan program studi yang ditawarkan seperti universitas, institusi, sekolah tinggi, politeknik, akademi dan akademi komunitas [2]. Mayoritas institusi pendidikan tinggi yang ada di Indonesia didominasi oleh sekolah tinggi dengan jumlah 2.364 kemudian diikuti oleh akademi 1.137, universitas sejumlah 505, politeknik berjumlah 225, dan institut berjumlah 110 [3]. Seperti terlihat pada Gambar 1.1.



Gambar 1.1 Jumlah perguruan tinggi di Indonesia [3]

Perguruan tinggi memiliki proses bisnis yang dijalankan di dalamnya, salah satunya yaitu kegiatan akademik. Kegiatan akademik sendiri memiliki beberapa *stakeholder* antara lain tenaga kependidikan, pendidik, dan peserta didik. Dalam menjalankan kegiatan tersebut, perguruan tinggi menggunakan sistem informasi akademik yang dapat mempermudah dalam integrasi data, penyajian data, dan sebagai pusat informasi serta alat rekam kegiatan akademik. Sistem informasi akademik harus mampu menangani semua proses bisnis yang dijalankan di perguruan tinggi tersebut. Salah satu proses bisnis yang akan diimplementasikan dalam Tugas Akhir ini adalah ekuivalensi.

Ekuivalensi adalah aktivitas memadankan mata kuliah atau konversi mata kuliah. Contoh kasus yang memiliki proses ekuivalensi yaitu saat pergantian kurikulum, dan penerimaan mahasiswa alih jenjang. Hasil dari proses ekuivalensi ini berupa daftar mata kuliah apa saja yang diakui dalam kurikulum yang baru. Di perguruan tinggi Indonesia, proses pergantian kurikulum merupakan hak perguruan tinggi masing-masing. Kerangka dasar dan struktur kurikulum perguruan tinggi dikembangkan oleh perguruan tinggi yang bersangkutan dengan mengacu pada standar nasional pendidikan untuk setiap program studi [4]. Secara otomatis jika terjadi perubahan pada kurikulum, maka proses ekuivalensi akan mengalami perubahan juga. Misalnya ekuivalensi dari kurikulum A ke kurikulum B akan berbeda dengan ekuivalensi kurikulum B ke kurikulum C. Contoh lain yang frekuensinya lebih sering terjadi yaitu pada proses penerimaan peserta didik alih jenjang. Proses ekuivalensi dilakukan untuk menyetarakan matakuliah yang sudah ditempuh terhadap matakuliah yang digunakan oleh perguruan tinggi yang baru. Jika masih dilakukan secara manual, proses ini akan menghabiskan banyak waktu dan tenaga karena memungkinkan sekali jika suatu perguruan tinggi menerima program alih jenjang dari banyak program studi dan perguruan tinggi lainnya. Dari beberapa kasus yang sudah disebutkan, dapat ditarik kesimpulan bahwa proses ekuivalensi

adalah salah satu proses yang penting, dan juga sangat rentan terhadap perubahan.

Sistem informasi akademik ini nantinya akan dibangun menggunakan bahasa pemrograman Java dan menggunakan basis data PostgreSQL. Dalam Tugas Akhir ini, penulis akan merancang perangkat lunak sistem informasi akademik yang pada modul ekuivalensi dengan basis web. Sistem informasi akademik yang generik sesuai kebutuhan umum perguruan tinggi bertujuan supaya sistem ini dapat digunakan oleh lebih dari satu perguruan tinggi.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut.

1. Bagaimana menangani perbedaan aturan ekuivalensi di setiap perguruan tinggi?
2. Bagaimana memadankan mata kuliah dari perguruan tinggi asal ke mata kuliah perguruan tinggi yang baru?
3. Bagaimana memadankan mata kuliah dari kurikulum lama ke kurikulum yang baru?
4. Bagaimana bentuk purwarupa sistem informasi akademik pada modul ekuivalensi?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, diantaranya adalah sebagai berikut.

1. Rancang bangun modul ekuivalensi ini berbasis web yang menggunakan bahasa pemrograman Java dan basis data PostgreSQL.
2. Rancang bangun sistem informasi akademik ini dilakukan pada modul ekuivalensi untuk kebutuhan jenjang pendidikan perguruan tinggi.

3. Modul ini hanya dapat diakses oleh 4 pengguna, yaitu administrator, tim ekuivalensi, tenaga kependidikan, dan peserta didik.
4. Modul ekuivalensi pada sistem ini membantu proses ekuivalensi saat alih jenjang, dan pergantian kurikulum.

1.4 Tujuan

Tujuan dari pengerjaan Tugas Akhir ini adalah sebagai berikut.

1. Membuat modul ekuivalensi yang dapat menangani perbedaan proses bisnis yang ada.
2. Membuat purwarupa modul ekuivalensi pada sistem informasi akademik yang memenuhi kebutuhan-kebutuhan secara generik.
3. Melakukan pengujian modul ekuivalensi.

1.5 Manfaat

Manfaat yang diharapkan dari Tugas Akhir ini adalah dapat membuat standar sistem informasi akademik berdasarkan kebutuhan yang umum modul ekuivalensi.

1.6 Metodologi

Tahapan-tahapan yang dilakukan dalam pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Penyusunan proposal Tugas Akhir.

Proposal Tugas Akhir ini berisi tentang pendahuluan yang meliputi deskripsi latar belakang, rumusan masalah, batasan masalah, tujuan dari pembuatan Tugas Akhir, dan manfaat dari Tugas Akhir. Selain itu dijelaskan juga tinjauan pustaka yang digunakan sebagai referensi pembuatan Tugas Akhir. Pada sub bab metodologi dijelaskan mengenai urutan

penyusunan Tugas Akhir dimulai dari pembuatan proposal sampai pembuatan buku Tugas Akhir.

2. Studi literatur

Studi literatur yang dilakukan berupa diskusi dan pemahaman mengenai topik Tugas Akhir ini, antara lain adalah sebagai berikut.

- a. Melakukan penggalian spesifikasi kebutuhan generik dari modul ekuivalensi pada sistem informasi akademik perguruan tinggi.
- b. Melakukan perancangan perangkat lunak yang sesuai kerangka kerja modularitas dan evolusi.

3. Survei dan wawancara

Tahap ini meliputi wawancara pada narasumber dari UNESA, UNAIR, UPN Veteran, UNNAR dan PENS. Wawancara dilakukan dengan memberikan tabel perbandingan antara analisis awal proses bisnis sistem informasi akademik modul ekuivalensi dan melihat persamaan atau perbedaan dengan proses bisnis sistem informasi akademik di perguruan tinggi lain.

4. Analisis dan desain perangkat lunak

Pengguna sistem informasi akademik modul ekuivalensi terdiri dari 4 aktor, yaitu administrator, tim ekuivalensi, tenaga kependidikan, dan peserta didik. Fitur yang dimiliki sistem ini yaitu:

1. mengelola katalog alih jenjang
2. mengelola matakuliah alih jenjang
3. membagikan katalog ke satuan manajemen
4. mengelola aturan ekuivalensi alih jenjang
5. mengelola peserta didik alih jenjang
6. mengelola ekuivalensi peserta didik alih jenjang
7. verifikasi matakuliah wajib
8. mencetak laporan ekuivalensi alih jenjang
9. mengelola aturan ekuivalensi
10. verifikasi ekuivalensi
11. mencetak laporan ekuivalensi

1.7 Sistematika Penulisan

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini:

Bab I Pendahuluan

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan tugas akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan tugas akhir.

Bab II Tinjauan Pustaka

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan tugas akhir ini.

Bab III Analisis dan Perancangan Sistem

Bab ini membahas mengenai desain dan perancangan perangkat lunak. Desain perangkat lunak meliputi desain data, arsitektur, dan proses.

Bab IV Implementasi

Bab ini membahas implementasi dari rancangan sistem yang dilakukan pada tahap perancangan.

Bab V Pengujian dan Evaluasi

Bab ini membahas pengujian dari aplikasi yang dibuat dengan melihat output yang dihasilkan oleh aplikasi, dan evaluasi untuk mengetahui kemampuan aplikasi.

Bab VI Kesimpulan dan Saran

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan serta saran-saran untuk pengembangan aplikasi pada masa mendatang.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan tugas akhir.

Lampiran

Merupakan bab tambahan yang berisi kode-kode sumber yang penting pada aplikasi ini.

BAB II

TINJAUAN PUSTAKA

Bab ini berisi penjelasan teori-teori yang berkaitan dengan rancangan modularitas dan evolusi pada modul ekuivalensi yang diajukan pada pengimplementasian program.

2.1 Penelitian Terkait

Penelitian-penelitian yang menjadi basis dari pengerjaan Tugas Akhir ini adalah Tugas Akhir yang mengacu pada proses rekayasa ulang dari sistem informasi akademik ITS [5] [6]. Purwarupa sistem informasi akademik yang dibangun pada penelitian-penelitian tersebut menggunakan ISO 9126 sebagai dasar dalam mengukur kualitasnya. ISO 9126 mendefinisikan kualitas produk perangkat lunak, model, karakteristik mutu, dan metriks terkait yang digunakan untuk mengevaluasi dan menetapkan kualitas sebuah produk perangkat lunak.

Pada penelitian-penelitian sebelumnya, perhatian utama dari pembangunan purwarupa sistem informasi akademik tersebut adalah karakteristik pemeliharaan [5] dan fungsionalitas [6]. Penelitian-penelitian tersebut mencoba memberikan solusi terhadap sistem informasi akademik ITS yang dinilai sulit untuk dipelihara dengan hasil akhir berupa purwarupa sistem informasi akademik. Namun, purwarupa-purwarupa dihasilkan tersebut tidak dirancang untuk bisa bekerja satu sama lain sehingga peningkatan mutu yang bisa tercapai hanya bisa dirasakan oleh sistem informasi akademik ITS saja.

Proposal tugas akhir ini mencoba untuk melakukan peningkatan terhadap penelitian-penelitian tersebut dengan menghasilkan sebuah purwarupa yang bisa digunakan oleh perguruan-perguruan tinggi secara umum. Salah satu penelitian tersebut [6] mencoba menggali kebutuhan fungsional terhadap proses penilaian dalam sistem informasi akademik ITS.

2.2 Perangkat Lunak Generik

Perangkat lunak generik, memiliki pengertian jenis perangkat lunak yang dapat digunakan untuk tujuan-tujuan yang berbeda tanpa harus melakukan modifikasi yang dibutuhkan. Contohnya adalah aplikasi *spreadsheet*, aplikasi ini dikatakan generik dikarenakan adanya *off-the-shelf package* yang menyediakan bagian akuntansi untuk keuangan, perencanaan pada penghitungan, perancangan pada *forecasting* dan lainnya tanpa adanya modifikasi [11]. Keuntungan menggunakan perangkat lunak generik adalah biaya yang lebih murah karena tidak dibutuhkannya pengembang untuk mengembangkan perangkat lunak secara khusus, aplikasi ini dapat langsung didapatkan dengan singkat karena tanpa adanya proses pembentukan aplikasi secara spesifik, kemudahan dalam pengembangan perangkat lunak dikarenakan pengembang akan terus-menerus mengembangkan aplikasi, dan menawarkan versi selanjutnya [12].

Pada sistem informasi akademik yang akan dibangun, generik yang dimaksud adalah memiliki proses bisnis yang generik, agar sistem informasi akademik ini dapat diterapkan di lebih dari satu universitas.

2.3 Ekuivalensi

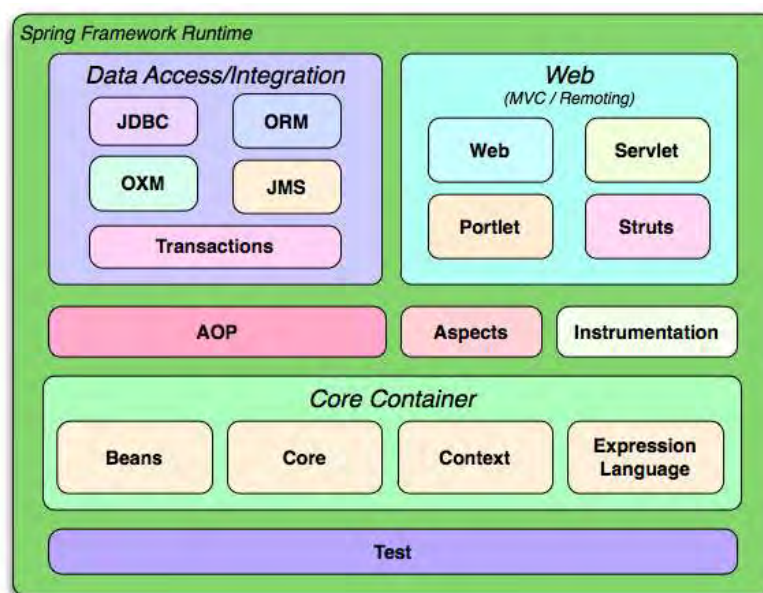
Ekuivalensi adalah aktivitas memadankan mata kuliah atau konversi mata kuliah. Contoh kasus yang memiliki proses ekuivalensi yaitu saat pergantian kurikulum, dan alih jenjang. Hasil dari proses ekuivalensi ini berupa daftar mata kuliah apa saja yang diakui dalam kurikulum yang baru. Di perguruan tinggi di Indonesia, proses pergantian kurikulum merupakan hak perguruan tinggi masing-masing. Terdapat perguruan tinggi yang pergantian kurikulumnya 3 tahun atau 5 tahun sekali. Kerangka dasar dan struktur kurikulum perguruan tinggi dikembangkan oleh perguruan tinggi yang bersangkutan dengan mengacu pada standar nasional pendidikan untuk setiap program studi [4]. Secara otomatis jika

terjadi perubahan pada kurikulum, maka proses ekuivalensi akan mengalami perubahan juga. Misalnya ekuivalensi dari kurikulum A ke kurikulum B akan berbeda dengan ekuivalensi kurikulum B ke kurikulum C. Sehingga, dapat ditarik kesimpulan jika proses ekuivalensi adalah salah satu proses yang sangat rentan terhadap perubahan.

Berdasarkan uraian di atas, lingkup modul ekuivalensi meliputi berbagai proses ekuivalensi saat terjadi pergantian kurikulum dan penerimaan peserta didik alih jenjang.

2.4 Kerangka Kerja Spring

Spring adalah salah satu kerangka kerja aplikasi untuk aplikasi berbasis Java atau sering disebut dengan JEE (*Java Enterprise Edition*) [7]. Spring menangani infrastruktur sistem, sehingga kita dapat berfokus pada lapisan aplikasi. Kerangka kerja Spring terdiri dari fitur-fitur yang diatur hingga 20 modul. Modul-modul ini dikelompokkan menjadi *Core Container*, *Data Access/Integration*, *Web*, *AOP* (*Aspect Oriented Programming*), *Instrumentation* dan *Test* [8].

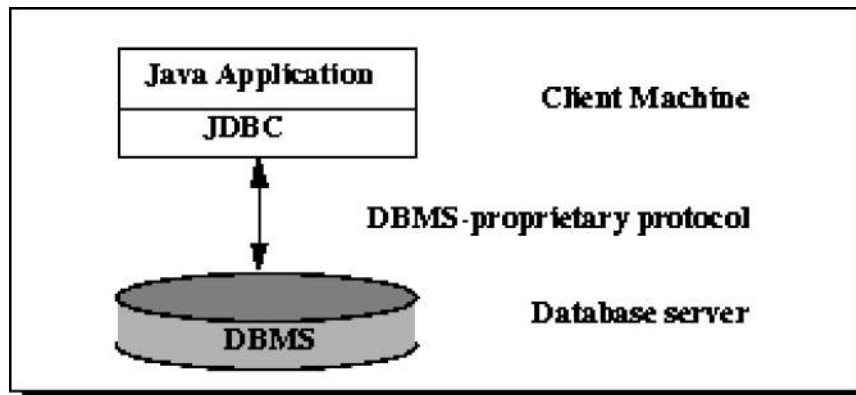


Gambar 2.1 Gambaran umum kerangka kerja spring [8]

Dalam pengerjaan Tugas Akhir ini, penggunaan modul kerangka kerja Spring yang ditunjukkan pada Gambar 2.1 adalah JDBC dan *Transaction* pada akses data/integrasi, Web-Servlet, Beans dan Context pada *Core Container*.

2.4.1 JDBC (*Java Database Connectivity*) [11]

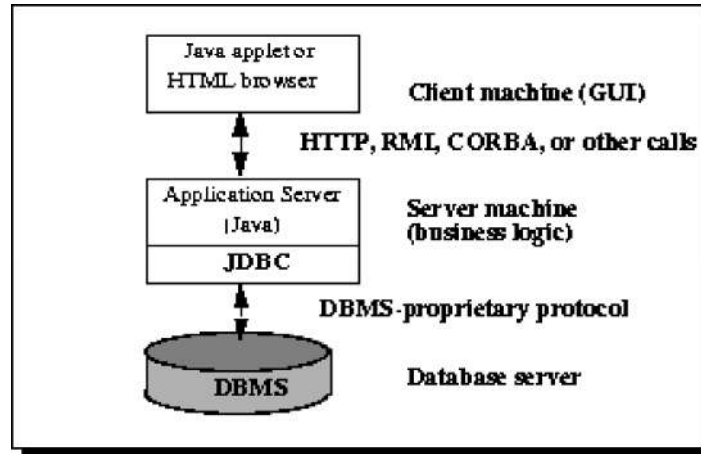
JDBC API merupakan Java API yang mampu mengakses beragam jenis data tabular, terutama data yang tersimpan dalam basis data relasional. Kemampuan JDBC terdiri dari 3 hal, yaitu menyambungkan sumber data dari basis data, mengirimkan *query* dan perubahan pada basis data, melakukan pengolahan dan temu kembali data yang diterima dari basis data untuk *query* yang dikirimkan pengguna. Arsitektur JDBC mendukung proses model untuk arsitektur dua lapisan (*two-tier*) dan tiga lapisan (*three-tier*) untuk akses ke basis data. Pada dan Gambar 2.3 akan dijelaskan mengenai visualisasi arsitektur dari JDBC.



Gambar 2.2 Arsitektur JDBC pada arsitektur *Two-Tier* [11]

Pada Gambar 2.2 aplikasi Java atau biasa disebut dengan Java Applet melakukan penyampaian data langsung pada sumber data. Tentunya ini dibutuhkan JDBC yang dapat mengakses kebutuhan aplikasi terhadap data di sumber data. Pengguna mengirimkan *query* pada basis data atau sumber data lainnya dan hasil data akan dikirimkan kembali pada pengguna. Sumber data

pada arsitektur ini ditempatkan pada mesin lainnya yang terkoneksi dengan jaringan internet.



Gambar 2.3 Arsitektur JDBC pada arsitektur *Three-Tier* [11]

Berbeda dengan arsitektur *three-tier* pada Gambar 2.3, perintah-perintah akan dikirimkan pada “*middle tier*” dari *service*, kemudian perintah-perintah tersebut diteruskan menuju sumber data. Sumber data akan melakukan pengolahan perintah dan mengirimkan permintaan data kembali pada “*middle tier*” dan dikirimkan ke pengguna.

2.4.2 *Spring-Transaction Manager* [10]

Sebuah transaksi basis data terdiri dari urutan aksi yang dianggap sebagai sebuah satuan pekerjaan dalam Spring. Aksi-aksi tersebut harus dapat berjalan seluruhnya atau tidak. Manajemen transaksi menjadi hal terpenting untuk aplikasi *enterprise* berorientasi RDBMS agar data dapat dijamin kebenarannya dan konsisten.

Konsep dari transaksi dijelaskan dengan 4 komponen yang sering disingkat menjadi ACID (*Atomicity, Consistency, Isolation, Durability*). *Atomicity* memiliki pengertian transaksi merupakan sebuah unit satuan pekerjaan walaupun dalam keberjalanannya berhasil atau tidak berhasil. *Consistency* dalam hal ini memberikan pengertian referensi pada basis data, seperti *primary key* pada tabel

dan lainnya. *Isolation* dibutuhkan apabila secara bersamaan terjadi proses transaksi, setiap transaksi harus terisolasi masing-masing untuk mencegah rusaknya data. *Durability* ialah setiap sebuah transaksi berhasil diselesaikan, hasil dari transaksi tersebut menjadi permanen dan tidak dapat dihapus dari basis data jika sistem mengalami kerusakan.

Penerapan 4 komponen dalam basis data dengan menggunakan SQL, secara mudah dapat dijelaskan menjadi 3 poin, yaitu sebagai berikut.

- Transaksi dimulai dengan menggunakan perintah *begin transaction*.
- Menjalankan aksi hapus, perbaharui atau tambah data menggunakan *query* SQL.
- Jika seluruh operasi berhasil, maka akan menjalankan perintah *commit*. Apabila operasi tidak berhasil, maka transaksi akan melakukan *rollback* pada seluruh operasi.

Pada kerangka kerja Spring terdapat *abstract layer* diatas API manajemen transaksi yang berbeda-beda. Tujuan adanya transaksi pada Spring adalah untuk menyediakan alternatif pada transaksi EJB (*Enterprise Java Beans*) dengan menambahkan kemampuan transaksi pada POJO (*Plain Old Java Object*). EJB membutuhkan server aplikasi, sedangkan manajemen transaksi pada Spring dapat diimplementasikan tanpa membutuhkan server aplikasi.

2.4.3 Servlet [13]

Servlet merupakan kelas dalam bahasa pemrograman Java yang digunakan untuk memperluas kemampuan server yang diakses oleh *host application*, dengan cara model pemrograman *request-response*. Walaupun *servlet* dapat merespon berbagai tipe *request*, biasanya *servlet* digunakan untuk memperluas aplikasi *host* di *web server*. Pada beberapa aplikasi, teknologi Java Servlet mendefinisikan kelas *servlet HTTP-Specific*.

Paket “*javax.servlet*” dan “*javax.servlet.http*” menyediakan *interface* dan kelas-kelas untuk

mengimplementasikan *servlet*. Seluruh *servlet* harus mengimplementasikan *interface servlet* yang sekaligus mendefinisikan metode *life-cycle*. Saat melakukan implementasi *service* yang sifatnya umum, dapat digunakan atau memperluas kelas “GenericServlet” yang menyediakan API Java Servlet. Pada kelas “HttpServlet” terdapat *method* seperti “doGet” dan “doPost” untuk mengelola *HTTP-specific service*.

2.4.4 Beans [11]

Kumpulan objek yang menjadi pembentuk aplikasi dan dikelola oleh wadah Spring IoC (*Inverse of Control*) sekarang bernama *Dependency Injection* adalah *beans*. Sebuah *bean* merupakan objek yang dipakai, dibentuk dan lainnya, dikelola oleh wadah Spring IoC. *Beans* ini dibuat dengan konfigurasi metadata yang diberikan pada wadah. Sebagai contoh, pada *file* XML terdapat definisi “<bean/>” yang berisikan informasi disebut metadata konfigurasi. Metadata konfigurasi dibutuhkan untuk mengetahui bagaimana pembuatan sebuah *bean*, rincian *lifecycle* *bean* dan *dependency* dari *bean*.

Tiga komponen *method* penting untuk menyediakan metadata konfigurasi pada *Spring Container*, terdiri dari *file* konfigurasi berbasis XML, konfigurasi berbasis anotasi dan konfigurasi berbasis Java.

2.5 PostgreSQL [12]

PostgreSQL atau sering disebut Postgres merupakan salah satu dari sejumlah basis data besar yang menawarkan skalabilitas, keluwesan, dan kinerja yang tinggi. Penggunaannya begitu meluas di berbagai *platform* dan didukung oleh banyak bahasa pemrograman. Bagi masyarakat TI (teknologi informasi) di Indonesia, Postgres sudah digunakan untuk berbagai aplikasi seperti *web*, sistem kasir, dan sistem informasi besar lainnya. Ada banyak hal unik yang bisa kita temui dari basis data yang satu ini.

Niatan awal para *programmer*-nya adalah membuat suatu basis data yang kaya akan fitur dengan keluwesan yang tinggi.

Postgres menawarkan standar yang lebih baik. Dibalik masalah teknis tersebut, Postgres tersedia dalam bentuk kode sumber dan dapat diunduh tanpa pembebanan biaya. Tidak heran kalau Linux Award sempat menobatkan Postgres sebagai basis data pilihan yang diikuti Oracle sebagai *runner-up*-nya.

Perbedaan penting antara Postgres dengan sistem relasional standar adalah arsitektur Postgres yang memungkinkan user untuk mendefinisikan sendiri SQL-nya, terutama pada pembuatan fungsi atau biasa disebut sebagai *stored procedure*. Hal ini dimungkinkan karena informasi yang disimpan oleh Postgres bukan hanya tabel dan kolom, melainkan tipe, fungsi, metode akses, dan banyak lagi yang terkait dengan tabel dan kolom tersebut. Semuanya terhimpun dalam bentuk kelas yang bisa diubah pengguna. Arsitektur yang menggunakan kelas ini lazim disebut berorientasi objek. Karena Postgres bekerja dengan kelas, berarti Postgres lebih mudah dikembangkan di tingkat pengguna, dan anda bisa mendefinisikan sebuah tabel sebagai turunan dari tabel lain. Sebagai perbandingan bahwa sistem basis data konvensional hanya dapat diperluas dengan mengubah kode sumbernya atau menggunakan modul tambahan yang ditulis khusus oleh vendor, maka dengan Postgres memungkinkan pengguna untuk membuat sendiri berkas objek atau *shared library* yang dapat diterapkan untuk mendefinisikan tipe data, fungsi, bahkan bahasa yang baru. Dengan demikian Postgres memiliki dua kekuatan besar yaitu kode sumber dan arsitektur yang luwes, tentunya disamping fitur penting lainnya seperti dokumentasi yang lengkap, dan sebagainya. Selain itu Postgres juga didukung oleh banyak antarmuka-antarmuka ke berbagai bahasa pemrograman seperti C++, Java, Perl, PHP, Python dan Tcl. ODBC dan JDBC juga tersedia yang membuat Postgres lebih terbuka dan dapat diterapkan secara meluas.

BAB IV

IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi berdasarkan rancangan yang telah dijabarkan sebelumnya. Implementasi yang dijelaskan meliputi lingkungan pembangunan perangkat lunak, implementasi kelas-kelas, implementasi proses bisnis ekuivalensi yang generik dan implementasi antarmuka perangkat lunak. Bahasa pemrograman yang digunakan untuk implementasi adalah bahasa pemrograman Java menggunakan kerangka kerja Spring.

4.1 Lingkungan Pembangunan Perangkat Lunak

Lingkungan pengembangan sistem yang digunakan untuk mengembangkan Tugas Akhir ini dilakukan pada lingkungan dan kaskas sebagai berikut.

1. Sistem operasi Windows 8.1 Professional 64 bit.
2. Spring Tool Suite 3.6.3 digunakan untuk kompilator.
3. Sybase Power Designer 15.0 merancang basis data.
4. StarUML digunakan untuk membuat diagram kasus penggunaan dan diagram aktivitas.
5. pgAdmin III digunakan untuk mengelola basis data.

4.2 Implementasi Kelas

Pada modul ekuivalensi, sistem yang dibuat mempunyai beberapa lapisan yang direpresentasikan dalam kelas. Lapisan tersebut adalah *view* sebagai lapisan antarmuka pengguna, *controller* sebagai tempat untuk menerima *request* yang dikirim oleh aplikasi *client* atau *browser*, *service* sebagai tempat pemrosesan data komputasi, *repository* sebagai tempat untuk

melakukan pengelolaan terhadap basis data dan *entity* sebagai representasi dari setiap tabel di basis data beserta relasinya.

4.2.1 Implementasi Kelas Entitas

Kelas entitas dalam kerangka kerja spring ditandai oleh anotasi *Entity* yang merepresentasikan tabel, kolom dan relasi di dalamnya. Kelas entitas ini nantinya akan dipakai oleh *Hibernate* untuk memetakan data menjadi objek.

Kelas entitas memetakan semua kolom-kolom tabel kepada *property* di dalam kelas termasuk relasi di dalamnya. Relasi yang dipetakan kelas entitas pada modul ekuivalensi hanya relasi *many to one* menggunakan tipe *fetch eager* dimana kelas yang berelasi dengan kelas tersebut akan di instansiasi secara bersamaan.

Strategi *eager* memiliki kelemahan yaitu pengambilan data yang dilakukan untuk menginstansiasi relasi belum tentu digunakan. Akan tetapi pada aplikasi berbasis *web* strategi *eager* memudahkan kelas *controller* dalam memetakan kelas entitas pada *string* JSON yang akan digunakan sebagai *response* pada kasus tertentu. Kode Sumber A.1 merupakan contoh dari kelas entitas yang merepresentasikan tabel katalog alih jenjang.

Pada modul ekuivalensi, kelas entitas yang digunakan sebanding dengan tabel yang dibutuhkan. Kelas entitas tersebut adalah sebagai berikut.

1. Kelas CalonPD

Merupakan kelas entitas yang digunakan untuk menyimpan peserta didik alih jenjang. Kelas ini digunakan pada kasus penggunaan UC-05.

2. Kelas KatalogAlihjenjang

Merupakan kelas entitas yang digunakan untuk menyimpan katalog alih jenjang. Kelas ini digunakan pada kasus penggunaan UC-01.

3. Kelas KatalogSatMan
Merupakan kelas entitas yang digunakan untuk memetakan akses katalog ke satman. Kelas ini digunakan pada kasus penggunaan UC-03.
4. Kelas MKAlihjenjang
Merupakan kelas entitas yang digunakan untuk menyimpan matakuliah alih jenjang. Kelas ini digunakan pada kasus penggunaan UC-02.
5. Kelas KRSCalonPD
Merupakan kelas entitas yang digunakan untuk menyimpan daftar matakuliah tempuh peserta didik alih jenjang. Kelas ini digunakan pada kasus penggunaan UC-06.
6. Kelas RelasiMKAlihjenjang
Merupakan kelas entitas yang digunakan untuk memetakan relasi dari tiap-tiap aturan ekuivalensi alih jenjang. Kelas ini digunakan pada kasus penggunaan UC-04.
7. Kelas EkuivalensiMKAlihjenjang
Merupakan kelas entitas yang digunakan untuk memetakan matakuliah alih jenjang dan matakuliah satuan manajemen yang saling berelasi. Kelas ini digunakan pada kasus penggunaan UC-04.
8. Kelas MKWajibCalonPD
Merupakan kelas entitas yang digunakan untuk menyimpan daftar matakuliah yang wajib ditempuh oleh peserta didik alih jenjang. Kelas ini digunakan pada kasus penggunaan UC-07.
9. Kelas AlihjenjangMKTerakui
Merupakan kelas entitas yang digunakan untuk menyimpan daftar matakuliah satuan manajemen yang terekuivalen saat proses ekuivalensi. Kelas ini digunakan pada kasus penggunaan UC-06.
10. Kelas RelasiEkuivalensi
Merupakan kelas entitas yang digunakan untuk memetakan relasi dari tiap-tiap aturan ekuivalensi kurikulum. Kelas ini digunakan pada kasus penggunaan UC-09.

11. Kelas EkuivalensiMK

Merupakan kelas entitas yang digunakan untuk memetakan matakuliah satuan manajemen yang saling berelasi saat ekuivalensi. Kelas ini digunakan pada kasus penggunaan UC-09.

12. Kelas EkuivalensiPD

Merupakan kelas entitas yang digunakan untuk menyimpan riwayat peserta didik yang sudah melalui proses ekuivalensi. Kelas ini digunakan pada kasus penggunaan UC-10.

13. Kelas MKWajibPD

Merupakan kelas entitas yang digunakan untuk menyimpan matakuliah wajib untuk peserta didik setelah proses ekuivalensi. Kelas ini digunakan pada kasus penggunaan UC-10, dan UC-11.

14. Kelas KRSHapus

Merupakan kelas entitas yang digunakan untuk menyimpan kartu rancangan studi yang dihapus saat proses ekuivalensi. Kelas ini digunakan pada kasus penggunaan UC-10.

4.2.2 Implementasi Kelas *Repository*

Kelas *repository* merupakan kelas yang digunakan untuk mengelola data yang meliputi mengambil, memasukan, mengubah, dan menghapus. Koneksi ke basis data dalam kelas ini ditangani oleh *session factory*. Pada modul ekuivalensi, terdapat kelas *repository* untuk setiap kelas entitas. Untuk setiap kelas *repository* pada modul ekuivalensi memiliki beberapa fungsi umum yang akan dijelaskan pada Tabel 4.1.

Kode Sumber A.2 merupakan contoh dari kelas *repository*. Kelas objek yang dijadikan sebagai parameter dan *return value* pada tabel Tabel 4.1 akan digantikan dengan kelas entitas masing masing.

Tabel 4.1 Fungsi umum kelas *repository*

Return value	Nama fungsi dan penjelasan
List<Obj>	get(String where, String order, int limit, int offset)
	Mengambil data pada tabel dengan filter where, urutan order batasan jumlah data limit dan mulai dari baris offset.
Obj	getById(UUID idTabel)
	Mengambil data dari tabel dengan id yang sama dengan idTabel.
UUID	insert(Obj data)
	Menyimpan atau mengubah data pada tabel dan mengembalikan id data.
void	update(Obj data)
	Mengubah data pada tabel..
void	delete(Obj obj)
	Menghapus data pada tabel
Long	count(String where)
	Menghitung data pada tabel dengan filter where

4.2.3 Implementasi Kelas *Service*

Kelas *service* merupakan kelas yang berfungsi sebagai pemroses data dan komputasi. Data yang diambil dari kelas *repository* diproses sesuai kasus penggunaan. Setiap kelas *repository* memiliki kelas *service*. Kelas ini sebagai penghubung antara kelas *controller* dan kelas *repository*. Untuk setiap kelas *service* memiliki beberapa fungsi umum yang akan dijelaskan pada Tabel 4.2 dan *property* umum yang akan dijelaskan pada Tabel 4.3.

Kode Sumber A.3 merupakan contoh dari kelas *service*. Kelas objek yang dijadikan sebagai parameter dan *return value* pada Tabel 4.1 akan digantikan dengan kelas entitas masing masing.

Tabel 4.2 Fungsi umum kelas *service*

Return value	Nama fungsi dan penjelasan
List<Obj>	get()
	Mengambil semua data pada tabel.
List<Obj>	get(String where)
	Mengambil data pada tabel dengan filter where.
List<Obj>	get(String where, String order)
	Mengambil data pada tabel dengan filter where dan urutan order.
List<Obj>	get(String where, String order, int limit, int offset)
	Mengambil data pada tabel dengan filter where, urutan order batasan jumlah data limit dan mulai dari baris offset.
Obj	getById(UUID idTabel)
	Mengambil data dari tabel dengan id yang sama dengan idTabel.
String	save(Obj data)
	Menyimpan atau mengubah data pada tabel dan mengembalikan id data.
String	delete(UUID idTabel)
	Menghapus data dari tabel dengan id yang sama dengan idTabel dan mengembalikan status proses penghapusan data.
Datatable	getdatatable(String sEcho, int iDisplayLength, int iDisplayStart, int iSortCol_0, String sSearch, String filter)
	Mengembalikan kelas datatable berisi data yang diminta berdasarkan banyak data iDisplayLength, awal data iDisplayStart, urutan data iSortCol_0, pencarian data berdasarkan kata kunci sSearch dan penyaringan data filter.

Tabel 4.3 *Property* umum kelas *service*

Tipe data	Nama <i>Property</i>
String[]	Column
	Merupakan kolom yang ditampilkan pada antarmuka yang memiliki komponen tabel master
Boolean[]	searchable(String where)
	Merupakan status apakah pencarian berdasarkan kata kunci bisa dilakukan untuk setiap kolom yang ditampilkan pada antarmuka yang memiliki komponen tabel master

4.2.4 Implementasi Kelas *Controller*

Kelas *controller* adalah kelas yang menangkap *request* dan data dari *client* lalu memberikan *response* berupa antarmuka atau string JSON dengan memproses terlebih dahulu di dalam kelas *controller* itu sendiri atau pada kelas *service*. Setiap fungsi yang ada di dalam kelas *controller* memiliki kode yang berbeda sesuai masing masing kasus penggunaan.

Pada kelas *controller* untuk setiap *request* yang meminta *response* berupa string JSON digunakan kelas *AjaxResponse* untuk menyeragamkan string JSON itu sendiri. Kelas *AjaxResponse* dapat dilihat pada Kode Sumber A.4.

Kelas *AjaxResponse* memiliki 3 *property* yaitu *String* status, *String* message dan *Object* data. Status merupakan status *request* yang diminta oleh klien, *message* adalah pesan terhadap *request* yang diminta, sedangkan data adalah data yang dikirim ke klien jika *request* tersebut memerlukan data.

4.3 Implementasi Proses Bisnis

Berikut merupakan implementasi dari analisa yang telah dipaparkan pada bab sebelumnya yaitu bagaimana cara membuat sistem informasi akademik generik yang dapat menangani variasi proses bisnis pada perguruan tinggi yang berbeda.

Untuk membuat sistem yang dinamis maka segala macam aturan seperti aturan ekuivalensi, pemetaan antar matakuliah, serta kemampuan aktor untuk menjalankan proses bisnis tertentu harus disimpan di dalam basis data sehingga perguruan tinggi tidak perlu merombak kode dalam sistem akademik jika terjadi perubahan aturan.

4.3.1 Mengelola Matakuliah Alih Jenjang

Pada proses ekuivalensi alih jenjang, dibutuhkan data matakuliah yang dimiliki oleh program studi asal peserta didik alih jenjang, dengan begitu matakuliah dapat dipetakan dan dibuat aturan ekuivalensinya. Pada umumnya, peserta didik alih jenjang akan menyertakan katalog matakuliah yang berisi detail kurikulum dan matakuliah pada program studi asalnya.

Pada kasus pengelolaan matakuliah alih jenjang terdapat kelas *service* yang dapat menangani proses pengelolaan matakuliah alih jenjang. Kode Sumber A.3 menjelaskan bagaimana cara mengelola matakuliah tersebut.

4.3.2 Pemetaan Matakuliah

Pada kasus pengelolaan aturan ekuivalensi, terdapat proses pemetaan matakuliah. Kode Sumber A.5 menjelaskan tentang fungsi menyimpan pemetaan matakuliah.

4.3.3 Pemrosesan aturan ekuivalensi

Pada proses pemetaan matakuliah, sistem menyediakan beberapa operator logika untuk mengatasi variasi aturan yaitu AND dan OR. Pemrosesan dilakukan dengan metode *infix to postfix*. Kode Sumber A.6 menjelaskan implementasi metode *infix to postfix*.

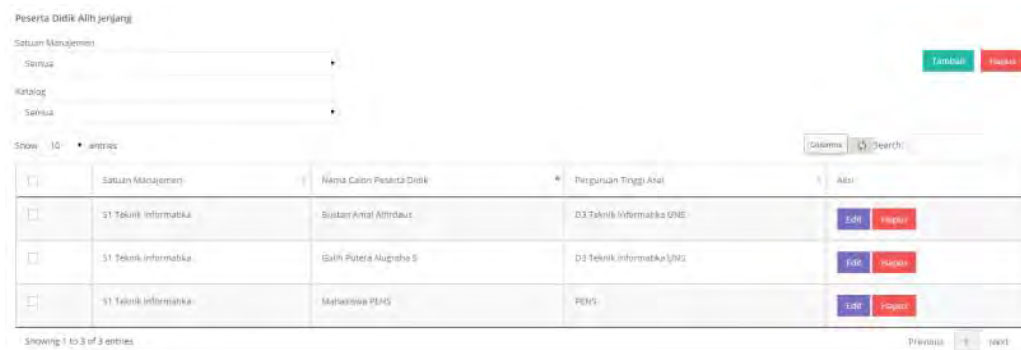
4.4 Implementasi Antarmuka Pengguna

Implementasi tampilan antarmuka pengguna pada program berbasis *web* dilakukan dengan menggunakan berkas JSP. Berikut

ini dijelaskan mengenai implementasi tampilan antarmuka pengguna yang terdapat pada modul ekuivalensi.

4.4.1 Implementasi Tabel Master

Tabel master merupakan antarmuka yang diimplementasikan pada halaman yang sudah dijelaskan pada 3.2.2.1. Implementasi tabel master dapat dilihat pada Gambar 4.1.



Peserta Didik Alih Jenjang

Satuan Manajemen: Semua

Kategori: Semua

Show 10 entries

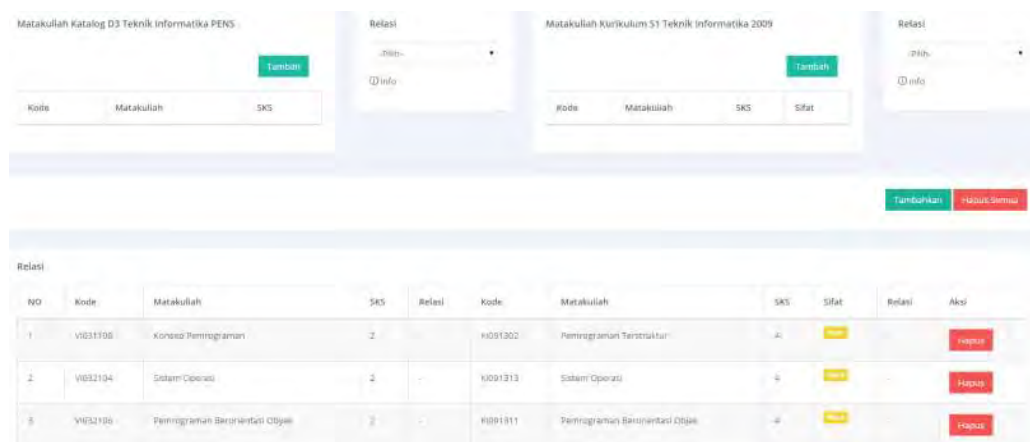
<input type="checkbox"/>	Satuan Manajemen	Nama Calon Peserta Didik	Pengurusan Tinggi Awal	Aksi
<input type="checkbox"/>	S1 Teknik Informatika	Budianto Amal Althadus	D3 Teknik Informatika GNE	Edit Hapus
<input type="checkbox"/>	S1 Teknik Informatika	Gahri Puteri Nugroho S	D3 Teknik Informatika UN2	Edit Hapus
<input type="checkbox"/>	S1 Teknik Informatika	Mahasiswa PENS	PENS	Edit Hapus

Showing 1 to 3 of 3 entries

Gambar 4.1 Implementasi antarmuka tabel master

4.4.2 Halaman Pemetaan Matakuliah

Halaman ini merupakan halaman yang digunakan untuk memetakan matakuliah. Pada terlihat bahwa beberapa matakuliah sudah terpetakan. Implementasi halaman pemetaan matakuliah dapat dilihat pada Gambar 4.2.



Matakuliah Katalog D3 Teknik Informatika PENS

Relasi: [Info](#)

Kode: Matakuliah: SKS

Matakuliah Kurikulum S1 Teknik Informatika 2009

Relasi: [Info](#)

Kode: Matakuliah: SKS: Sifat

Tambahkan: Hapus Semua

Relasi

NO	Kode	Matakuliah	SKS	Relasi	Kode	Matakuliah	SKS	Sifat	Relasi	Aksi
1	V031106	Konsep Pemrograman	2	-	H091302	Pemrograman Terstruktur	4	Prasyarat	-	Hapus
2	V032104	Sistem Operasi	2	-	H091313	Sistem Operasi	4	Prasyarat	-	Hapus
3	V032106	Pemrograman Berorientasi Objek	2	-	H091311	Pemrograman Berorientasi Objek	4	Prasyarat	-	Hapus

Gambar 4.2 Implementasi antarmuka halaman pemetaan matakuliah

4.4.3 Implementasi *Form Detail*

Form Detail merupakan antarmuka yang diimplementasikan pada halaman yang sudah disebutkan pada 3.2.2.2. Implementasi *form detail* dapat dilihat pada Gambar 4.3.

Kelola Calon Peserta Didik

Satuan Manajemen
S1 Teknik Informatika

☒ Lakukan proses penyetaraan matakuliah

Katalog
Katalog D3 Teknik Informatika PENS

Nama Calon Peserta Didik
Bustan Amal Alfirdaus

Tempat Lahir
Pati

Tanggal Lahir
29-08-1993

Alamat
Pati

Perguruan Tinggi Asal
D3 Teknik Informatika UNS

Simpan Kembali

Gambar 4.3 Implementasi antarmuka *form detail*

4.4.4 Halaman Proses Ekuivalensi Alih Jenjang

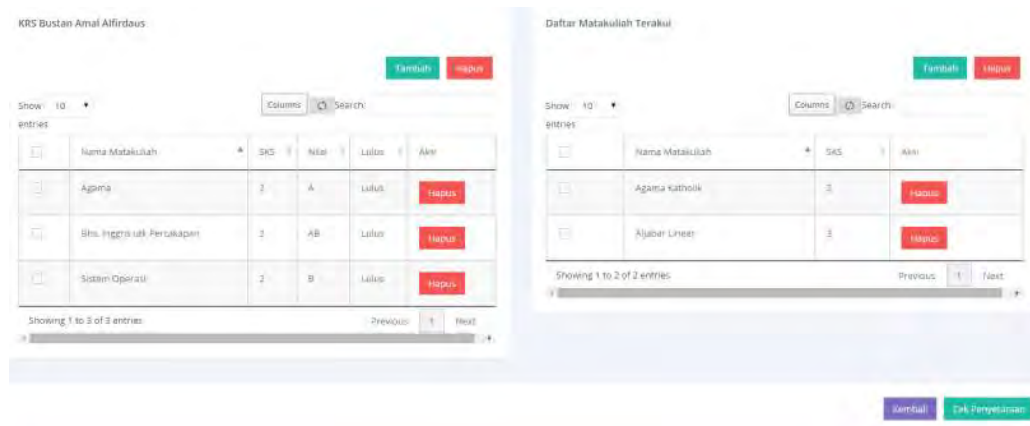
Halaman ini merupakan halaman yang digunakan untuk proses ekuivalensi. Pada Gambar 4.4 terlihat beberapa matakuliah yang sudah diakui.

4.4.5 Halaman Kelola Matakuliah Wajib

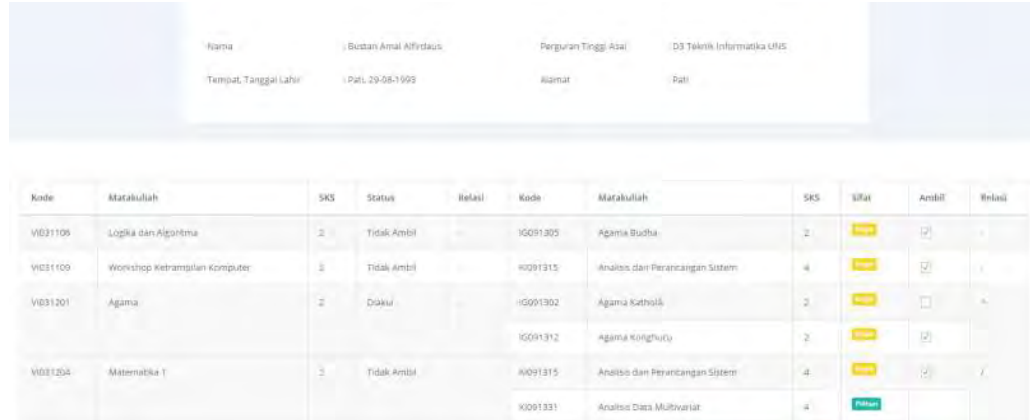
Halaman ini merupakan halaman yang digunakan untuk mengelola matakuliah wajib untuk peserta didik alih jenjang. Pada Gambar 4.5 terlihat bahwa beberapa matakuliah dari katalog sudah terakui dan terdapat juga matakuliah yang wajib diambil oleh peserta didik alih jenjang.

4.4.6 Halaman Proses Ekuivalensi Kurikulum

Halaman ini merupakan halaman yang digunakan untuk proses ekuivalensi kurikulum. Pada Gambar 4.6 terlihat dua daftar matakuliah dari kurikulum yang berbeda.



Gambar 4.4 Implementasi antarmuka halaman ekuivalensi alih jenjang



Gambar 4.5 Implementasi antarmuka halaman kelola matakuliah wajib

Kode	Matakuliah	SKS	Sifat	Status	Hapus
K091301	Aljabar Linear	3	Teori	Selesai	<input type="checkbox"/>
K091308	Bhs Inggris	2	Teori	Belum Selesai	<input type="checkbox"/>
SA091201	Kalkulus I	3	Teori	Belum Selesai	<input type="checkbox"/>
K091302	Penrograman Terstruktur	4	Teori	Selesai	<input type="checkbox"/>
K091303	Sistem dan Teknologi Informasi	2	Teori	Selesai	<input type="checkbox"/>
K091304	Sistem Digital	4	Teori	Selesai	<input type="checkbox"/>
K091305	Agama Budha	2	Teori	Belum Selesai	<input type="checkbox"/>
K091304	Agama Hindu	2	Teori	Belum Selesai	<input type="checkbox"/>
K091301	Agama Islam	2	Teori	Belum Selesai	<input type="checkbox"/>
K091302	Agama Katolik	2	Teori	Belum Selesai	<input type="checkbox"/>

Kode	Matakuliah	SKS	Sifat	Status	Ambil
SM141203	Kalkulus I	3	Teori	Selesai	<input type="checkbox"/>
K141331	Matematika Informatika	3	Teori	Selesai	<input type="checkbox"/>
K141301	Dasar Pemrograman	4	Teori	Selesai	<input type="checkbox"/>
K041108	Bahasa Inggris	3	Teori	Selesai	<input checked="" type="checkbox"/>
K141311	Sistem Basis Data	4	Teori	Selesai	<input checked="" type="checkbox"/>
K141312	Sistem Operasi	4	Teori	Selesai	<input checked="" type="checkbox"/>
K141403	Arsitektur Perangkat Lunak	3	Praktik	Selesai	<input type="checkbox"/>
K141438	Analisis Media Sosial	3	Praktik	Selesai	<input type="checkbox"/>
K141302	Matematika Disant	3	Teori	Selesai	<input checked="" type="checkbox"/>
K141303	Sistem dan Teknologi Informasi	2	Teori	Selesai	<input checked="" type="checkbox"/>

Gambar 4.6 Implementasi antarmuka halaman proses ekuivalensi kurikulum

BAB V

PENGUJIAN DAN EVALUASI

Pada bab ini akan dijelaskan uji coba yang dilakukan pada modul ekuivalensi. Pembahasan pengujian meliputi lingkungan uji coba dan skenario uji coba yang merupakan uji fungsionalitas dan *usability* pada kasus penggunaan.

5.1 Lingkungan Uji Coba

Lingkungan pengujian adalah lingkungan, baik perangkat keras maupun perangkat lunak tempat pengujian sistem dilakukan. Pengujian untuk Tugas Akhir ini dilakukan pada komputer dengan spesifikasi sebagai berikut.

1. Perangkat keras komputer *server*
 - a. Prosesor: Intel® Core™ i7-3630QM CPU @ 2.40GHz
 - b. *Memory*(RAM): 8,00 GB
 - c. Tipe sistem: 64-bit sistem operasi
2. Perangkat lunak komputer *server*
 - a. Sistem operasi: Windows 8.1 Professional.
 - b. Server aplikasi web: Pivotal tc Server Integration.
 - c. Manajemen basis data: PostgreSQL.
3. Perangkat lunak komputer *client*
 - a. *Web browser*: Google Chrome.

5.2 Skenario Pengujian

Pada sub bab ini dijelaskan skenario pengujian yang digunakan. Pengujian yang dilakukan menggunakan metode pengujian *control testing*. *Control testing* merupakan pengujian aplikasi yang terkendali. Pengguna melakukan pengujian fungsionalitas aplikasi didampingi oleh pengembang dengan menggunakan skenario yang telah ditentukan oleh pengembang.

5.2.1 Pengujian Fungsionalitas

Pengujian fungsionalitas dilakukan oleh 2 orang yang berperan sebagai administrator dan tim ekuivalensi. Hasil pengujian berupa rekap kuisioner. Rekap kuisioner dibagi menjadi 3 bagian sesuai fungsionalitas aktor. Rekap fungsionalitas untuk kepala satuan manajemen pada Tabel 5.3, kepala tim ekuivalensi pada Tabel 5.1, peserta didik pada Tabel 5.2.

Tabel 5.1 Kebutuhan fungsional untuk tim ekuivalensi

No	Kebutuhan Fungsionalitas	Terpenuhi
1	Sistem memiliki kemampuan untuk mengelola ekuivalensi saat pergantian kurikulum	2
2	Sistem memiliki kemampuan untuk memetakan matakuliah pada kurikulum yang lama dengan matakuliah pada kurikulum yang baru	2
3	Sistem dapat mengelola persetujuan ekuivalensi pada tiap-tiap mahasiswa	2
4	Sistem dapat mengelola matakuliah yang wajib diambil pada kurikulum baru dan dapat menghapus krs lulus yang sudah pernah ditempuh	2
5	Sistem menyediakan laporan matakuliah yang wajib diambil mahasiswa alih jenjang setelah melakukan simpan permanen	2
6	Sistem dapat membatalkan simpan permanen sehingga data matakuliah yang wajib diambil masih bisa diubah	2

Tabel 5.2 Kebutuhan fungsional untuk peserta didik

No	Kebutuhan Fungsionalitas	Terpenuhi
1	Membuka laporan ekuivalensi yang sudah dilakukan mahasiswa	2

Tabel 5.3 Kebutuhan fungsional untuk kepala satuan manajemen

No	Kebutuhan Fungsionalitas	Terpe nuhi
1	Sistem harus memiliki kemampuan untuk mengelola katalog alih jenjang	2
2	Sistem harus memiliki kemampuan untuk mengelola matakuliah yang dimiliki katalog alih jenjang tersebut	2
3	Sistem memiliki kemampuan untuk mengelola ekuivalensi atau pemetaan matakuliah di katalog terhadap matakuliah yang dimiliki jurusan	2
4	Sistem menyediakan penjelasan mengenai apa itu relasi	2
5	Sistem harus memiliki kemampuan untuk mengelola mahasiswa alihjenjang	2
6	Sistem dapat memasukkan matakuliah yang sudah ditempuh oleh mahasiswa alihjenjang	2
7	Sistem dapat melakukan penyetaraan matakuliah sesuai pemetaan yang dilakukan sebelumnya	2
8	Sistem dapat mengelola matakuliah wajib bagi mahasiswa alihjenjang	2
9	Sistem menyediakan laporan matakuliah yang wajib diambil mahasiswa alihjenjang setelah melakukan simpan permanen	2
10	Sistem dapat membatalkan simpan permanen sehingga data matakuliah yang wajib diambil masih bisa diubah	2

Semua fungsionalitas yang dimiliki sistem dapat berfungsi dengan baik.

5.2.1.1 Pengujian fungsionalitas mengelola katalog alih jenjang

Tabel 5.4 Rincian pengujian fungsionalitas mengelola katalog alih jenjang

Nama Skenario Pengujian	Fungsionalitas mengelola katalog alih jenjang.
Kode	PF-001
Tujuan Pengujian	Menguji fungsi tambah katalog alih jenjang.
Aktor	Pengguna merupakan administrator.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu kelola katalog alih jenjang. 2. Pengguna menekan tombol tambah. 3. Pengguna mengisi formulir katalog alih jenjang. 4. Pengguna menekan tombol simpan.
Masukan	<ol style="list-style-type: none"> 1. Nama Katalog. 2. Catatan
Hasil yang diharapkan	Katalog baru tersimpan.
Hasil yang diperoleh	Katalog baru tersimpan.
Hasil pengujian	Berhasil.

5.2.1.2 Pengujian fungsionalitas mengelola matakuliah alih jenjang

Tabel 5.5 Rincian pengujian fungsionalitas mengelola matakuliah alih jenjang

Nama Skenario Pengujian	Fungsionalitas mengelola matakuliah alih jenjang.
Kode	PF-002
Tujuan Pengujian	Menguji fungsi tambah matakuliah alih jenjang.
Aktor	Pengguna merupakan administrator.

Tabel 5.6 Rincian pengujian fungsionalitas mengelola matakuliah alih jenjang (2)

Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu kelola matakuliah alih jenjang. 2. Pengguna menekan tombol tambah. 3. Pengguna mengisi formulir matakuliah alih jenjang. 4. Pengguna menekan tombol simpan.
Masukan	<ol style="list-style-type: none"> 1. Kode Matakuliah. 2. Nama Matakuliah. 3. Jumlah SKS. 4. Deskripsi
Hasil yang diharapkan	Matakuliah baru tersimpan.
Hasil yang diperoleh	Matakuliah baru tersimpan.
Hasil pengujian	Berhasil.

5.2.1.3 Pengujian fungsionalitas membagikan katalog ke satuan manajemen

Tabel 5.7 Rincian pengujian fungsionalitas membagikan katalog ke satuan manajemen

Nama Skenario Pengujian	Fungsionalitas membagikan katalog ke satuan manajemen
Kode	PF-003
Tujuan Pengujian	Menguji fungsi tambah hak akses satuan manajemen terhadap katalog
Aktor	Pengguna merupakan administrator.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu kelola katalog satman 2. Pengguna menekan tombol tambah. 3. Pengguna mengisi formulir katalog satman 4. Pengguna menekan tombol simpan.

Tabel 5.8 Rincian pengujian fungsionalitas membagikan katalog ke satuan manajemen (2)

Masukan	1. Katalog. 2. Satuan Manajmen
Hasil yang diharapkan	Katalog dapat diakses oleh satuan manajemen.
Hasil yang diperoleh	Katalog dapat diakses oleh satuan manajemen.
Hasil pengujian	Berhasil.

5.2.1.4 Pengujian fungsionalitas mengelola peserta didik alih jenjang

Tabel 5.9 Rincian pengujian fungsionalitas mengelola pesera didik alih jenjang

Nama Skenario Pengujian	Fungsionalitas mengelola peserta didik alih jenjang.
Kode	PF-004
Tujuan Pengujian	Menguji fungsi tambah peserta didik alih jenjang.
Aktor	Pengguna merupakan administrator.
Prosedur Pengujian	1. Pengguna membuka menu kelola peserta didik alih jenjang. 2. Pengguna menekan tombol tambah. 3. Pengguna mengisi formulir peserta didik alih jenjang. 4. Pengguna menekan tombol simpan.
Masukan	1. Satuan Manajemen. 2. Katalog 3. Nama Peserta Didik Alih Jenjang 4. Tempat Lahir 5. Tanggal Lahir 6. Alamat 7. Perguruan Tinggi Asal
Hasil yang diharapkan	Peserta didik baru tersimpan.

Tabel 5.10 Rincian pengujian fungsionalitas mengelola peserta didik alih jenjang (2)

Hasil yang diperoleh	Peserta didik baru tersimpan.
Hasil pengujian	Berhasil.

5.2.1.5 Pengujian fungsionalitas ekuivalensi peserta didik alih jenjang

Tabel 5.11 Rincian pengujian fungsionalitas ekuivalensi peserta didik alih jenjang

Nama Skenario Pengujian	Fungsionalitas ekuivalensi peserta didik alih jenjang
Kode	PF-005
Tujuan Pengujian	Menguji fungsi ekuivalensi alih jenjang
Aktor	Pengguna merupakan tim ekuivalensi.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu ekuivalensi alih jenjang 2. Pengguna menekan tombol kelola ekuivalensi 3. Pengguna memilih daftar ekuivalensi 4. Pengguna menekan tombol tambah matakuliah tempuh. 5. Pengguna menambah matakuliah tempuh. 6. Pengguna menekan tombol cek ekuivalensi
Masukan	<ol style="list-style-type: none"> 1. Daftar matakuliah tempuh peserta didik alih jenjang
Hasil yang diharapkan	Matakuliah berhasil disetarakan sesuai aturan ekuivalensi.
Hasil yang diperoleh	Matakuliah berhasil disetarakan sesuai aturan ekuivalensi.
Hasil pengujian	Berhasil.

5.2.1.6 Pengujian fungsionalitas verifikasi matakuliah wajib

Tabel 5.12 Rincian pengujian fungsionalitas verifikasi matakuliah wajib

Nama Skenario Pengujian	Fungsionalitas verifikasi matakuliah wajib
Kode	PF-006
Tujuan Pengujian	Menguji fungsi verifikasi matakuliah wajib
Aktor	Pengguna merupakan tim ekuivalensi.
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu ekuivalensi alih jenjang 2. Pengguna menekan tombol kelola mk wajib 3. Pengguna memilih kurikulum 4. Pengguna menentukan matakuliah wajib yang akan diambil 5. Pengguna menekan tombol simpan
Masukan	1. Daftar matakuliah wajib ambil tersimpan
Hasil yang diharapkan	Matakuliah wajib ambil tersimpan.
Hasil yang diperoleh	Matakuliah wajib ambil tersimpan.
Hasil pengujian	Berhasil.

5.2.1.7 Pengujian fungsionalitas mencetak laporan ekuivalensi alih jenjang

Tabel 5.13 Rincian pengujian fungsionalitas mencetak laporan ekuivalensi alih jenjang

Nama Skenario Pengujian	Fungsionalitas mencetak laporan ekuivalensi alih jenjang
Kode	PF-007
Tujuan Pengujian	Menguji fungsi mencetak laporan ekuivalensi alih jenjang
Aktor	Pengguna merupakan administrator

Tabel 5.14 Rincian pengujian fungsionalitas mencetak laporan ekuivalensi alih jenjang (2)

Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka laporan alih jenjang 2. Pengguna memilih satuan manajemen 3. Pengguna memilih peserta didik alih jenjang 4. Pengguna menekan tombol cetak
Masukan	1. Peserta didik alih jenjang
Hasil yang diharapkan	Muncul laporan versi cetak
Hasil yang diperoleh	Muncul laporan versi cetak
Hasil pengujian	Berhasil.

5.2.1.8 Pengujian fungsionalitas mencetak laporan ekuivalensi

Tabel 5.15 Rincian pengujian fungsionalitas mencetak laporan ekuivalensi

Nama Skenario Pengujian	Fungsionalitas mencetak laporan ekuivalensi
Kode	PF-008
Tujuan Pengujian	Menguji fungsi mencetak laporan ekuivalensi
Aktor	Pengguna merupakan tim kurikulum
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka laporan ekuivalensi 2. Pengguna memilih kurikulum 3. Pengguna memilih peserta didik 4. Pengguna menekan tombol cetak
Masukan	1. Peserta didik
Hasil yang diharapkan	Muncul laporan versi cetak
Hasil yang diperoleh	Muncul laporan versi cetak
Hasil pengujian	Berhasil.

5.2.1.9 Pengujian fungsionalitas verifikasi ekuivalensi

Tabel 5.16 Rincian pengujian fungsionalitas verifikasi ekuivalensi

Nama Skenario Pengujian	Fungsionalitas verifikasi ekuivalensi
Kode	PF-009
Tujuan Pengujian	Menguji fungsi verifikasi ekuivalensi kurikulum
Aktor	Pengguna merupakan tim ekuivalensi
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu persetujuan ekuivalensi 2. Pengguna menekan tombol tambah 3. Pengguna memilih ekuivalensi 4. Pengguna memilih peserta didik 5. Pengguna menentukan matakuliah wajib dan krs yang akan dihapus
Masukan	1. Matakuliah wajib dan krs yang dihapus
Hasil yang diharapkan	Matakuliah wajib dan krs yang dihapus berhasil disimpan.
Hasil yang diperoleh	Matakuliah wajib dan krs yang dihapus berhasil disimpan.
Hasil pengujian	Berhasil.

5.2.1.10 Pengujian fungsionalitas mengelola aturan ekuivalensi kurikulum

Tabel 5.17 Rincian pengujian fungsionalitas mengelola aturan ekuivalensi kurikulum

Nama Skenario Pengujian	Fungsionalitas mengelola aturan ekuivalensi kurikulum
Kode	PF-010
Tujuan Pengujian	Menguji fungsi mengelola aturan ekuivalensi
Aktor	Pengguna merupakan tim ekuivalensi

Tabel 5.18 Rincian pengujian fungsionalitas mengelola aturan ekuivalensi kurikulum (2)

Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka ekuivalensi kurikulum 2. Pengguna menekan tombol tambah 3. Pengguna memilih kurikulum 4. Pengguna menambahkan aturan ekuivalensi 5. Pengguna menyimpan aturan ekuivalensi
Masukan	1. Aturan ekuivalensi berhasil disimpan
Hasil yang diharapkan	Aturan ekuivalensi berhasil disimpan.
Hasil yang diperoleh	Aturan ekuivalensi berhasil disimpan.
Hasil pengujian	Berhasil.

5.2.1.11 Pengujian fungsionalitas mengelola aturan ekuivalensi alih jenjang

Tabel 5.19 Rincian pengujian fungsionalitas mengelola aturan ekuivalensi alih jenjang

Nama Skenario Pengujian	Fungsionalitas mengelola aturan ekuivalensi alih jenjang
Kode	PF-011
Tujuan Pengujian	Menguji fungsi mengelola aturan ekuivalensi alih jenjang
Aktor	Pengguna merupakan tim ekuivalensi
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka kelola katalog satman 2. Pengguna memilih katalog 3. Pengguna menekan tombol tambah 4. Pengguna memilih kurikulum 5. Pengguna menambahkan aturan ekuivalensi 6. Pengguna menyimpan aturan ekuivalensi
Masukan	1. Aturan ekuivalensi berhasil disimpan
Hasil yang diharapkan	Aturan ekuivalensi berhasil disimpan.
Hasil yang diperoleh	Aturan ekuivalensi berhasil disimpan.
Hasil pengujian	Berhasil.

5.2.2 Perbedaan Aturan Ekuivalensi

Di Universitas Narotama memiliki perbedaan aturan mengenai matakuliah yang memiliki prasyarat. Matakuliah yang memiliki prasyarat tersebut tidak otomatis terekuivalensi jika matakuliah prasyaratnya belum terekuivalensi juga. Karena perbedaan ini tidak dimiliki oleh perguruan tinggi yang lainnya, maka solusi yang diberikan yaitu dengan memberikan pemberitahuan pada saat membuat aturan ekuivalensi. Skenario pengujian yang dilakukan dapat dilihat pada Tabel 5.20 dan Tabel 5.21, dengan hasil yang dapat dilihat pada Gambar 5.1.

5.2.3 Proses Pemadanan Matakuliah

Proses pemadanan matakuliah dilakukan dengan cara memetakannya satu per satu. Satu matakuliah dapat dipadankan dengan banyak matakuliah dan juga sebaliknya. Skenario pengujian yang dilakukan dapat dilihat pada Tabel 5.22, dengan hasil yang dapat dilihat pada Gambar 5.2.

Tabel 5.20 Skenario matakuliah memiliki prasyarat

Nama Skenario Pengujian	Skenario matakuliah memiliki prasyarat
Kode	P-01
Tujuan Pengujian	Menguji fungsi pengecekan matakuliah yang memiliki prasyarat.
Kondisi Awal	-
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu kelola katalog satman. 2. Pengguna memilih katalog. 3. Pengguna memilih kurikulum. 4. Pengguna memasukkan aturan dengan matakuliah yang memiliki prasyarat
Masukan	-

Tabel 5.21 Skenario matakuliah memiliki prasyarat (2)

Hasil yang diharapkan	Pemberitahuan muncul.
Hasil yang diperoleh	Pemberitahuan muncul.
Hasil pengujian	Berhasil.

Tabel 5.22 Skenario pemadanan matakuliah

Nama Skenario Pengujian	Skenario pemadanan matakuliah
Kode	PF-02
Tujuan Pengujian	Menguji fungsi pemadanan matakuliah.
Kondisi Awal	-
Prosedur Pengujian	<ol style="list-style-type: none"> 1. Pengguna membuka menu kelola katalog satman. 2. Pengguna memilih katalog. 3. Pengguna memilih kurikulum. 4. Pengguna memasukkan matakuliah yang nantinya akan menjadi aturan-aturan. 5. Pengguna menyimpan aturan-aturan tersebut.
Masukan	-
Hasil yang diharapkan	Pemetaan matakuliah berhasil disimpan.
Hasil yang diperoleh	Pemetaan matakuliah berhasil disimpan.
Hasil pengujian	Berhasil.

Q **Matakuliah memiliki prasyarat**

Matakuliah Kurikulum S1 Teknik Informatika 2009

Tambah

Kode	Matakuliah	SKS	Sifat
KI091302	Pemrograman Terstruktur	4	Prasyat

Relasi

-Pilih-

info

Gambar 5.1 Hasil skenario matakuliah memiliki prasyarat

Relasi

NO	Kode	Matakuliah	SKS	Relasi	Kode	Matakuliah	SKS	Sifat	Relasi	Aksi
1	VI032104	Sistem Operasi	2	-	KI091313	Sistem Operasi	4	Wajib	-	Hapus
2	VI032106	Pemrograman Berorientasi Obyek	2	-	KI091311	Pemrograman Berorientasi Obyek	4	Wajib	-	Hapus
3	VI031108	Konsep Pemrograman	2	-	KI091302	Pemrograman Terstruktur	4	Wajib	-	Hapus

Gambar 5.2 Hasil skenario pemadanan matakuliah

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang diambil selama pengerjaan tugas akhir beserta saran-saran tentang pengembangan yang dapat dilakukan terhadap tugas akhir ini di masa yang akan datang.

6.1 Kesimpulan

Dari hasil pengamatan selama proses perancangan, implementasi dan uji coba perangkat lunak yang dilakukan, dapat diambil kesimpulan sebagai berikut.

1. Modul ekuivalensi dalam sistem informasi akademik dapat menangani perbedaan aturan ekuivalensi dengan cara menentukan proses bisnis yang baku.
2. Sistem dapat memadankan matakuliah dari perguruan tinggi asal ke matakuliah perguruan tinggi yang baru dengan cara menyimpan data matakuliah dari perguruan tinggi asal dan memetakannya terhadap terhadap matakuliah pada perguruan tinggi yang baru.
3. Sistem dapat memadankan matakuliah dari kurikulum lama terhadap kurikulum yang baru dengan cara melakukan pemetaan matakuliah di kurikulum yang lama terhadap matakuliah kurikulum baru.
4. Bentuk purwarupa sistem informasi akademik pada modul ekuivalensi seperti pada sistem yang telah dibangun.

6.2 Saran

Terdapat saran terkait Tugas Akhir ini yang diharapkan dapat membuat Tugas Akhir ini menjadi lebih baik. Saran tersebut yaitu diperlukan penelitian lebih lanjut terhadap perguruan tinggi yang menerapkan proses ekuivalensi agar data

variasi proses bisnis lebih banyak dan dapat dibuat rancangan proses bisnis yang lebih umum.

DAFTAR PUSTAKA

- [1] R. Indonesia, “Peraturan Pemerintah Republik Indonesia Nomor 60 Tahun 1999 Tentang Pendidikan Tinggi”. Indonesia 1999.
- [2] “Direktorat Jenderal Pendidikan Tinggi,” Kementerian Pendidikan dan Kebudayaan, [Online]. Available: <http://dikti.go.id/profil-dikti/sistem-pendidikan-tinggi/>. [Diakses 27 Desember 2014].
- [3] “Direktorat Jenderal Pendidikan Tinggi,” Kementerian Pendidikan dan Kebudayaan, [Online]. Available: <http://forlap.dikti.go.id/perguruan tinggi/homegraphpt>. [Diakses 27 Desember 2014].
- [4] R. Indonesia, “Undang-Undang Republik Indonesia Nomor 20 Tahun 2003”. Indonesia 2003.
- [5] Y. D. Nurseha, Rekayasa Ulang Penelitian Sistem Informasi Akademik ITS Berdasarkan Karakteristik Fungsionalitas Model Kualitas ISO/IEC 9126, Surabaya: ITS, 2014.
- [6] A. B. Raharjo, Rekayasa Ulang SIM Akademik ITS Berdasarkan Karakteristik Pemeliharaan Menggunakan Model Kualitas ISO/IEC 9126, Surabaya: Institut Teknologi Sepuluh Nopember, 2013.
- [7] “Teach-ICT.com,” [Online]. Available: <http://www.teach-ict.com/glossary/G/genericsoftware.htm>. [Diakses 9 Januari 2015].
- [8] “Teach-ICT.com,” [Online]. Available: http://www.teach-ict.com/as_a2_ict_new/ocr/A2_G063/335_implementing_systems/implement_custom/miniweb/pg4.htm. [Diakses 9 Januari 2015].
- [9] M. TOHA, IMPLEMENTASI FRAMEWORK SPRING MVC UNTUK PEMBUATAN SISTEM INFORMASI

MANAJEMEN E-COMMERCE, UNIVERSITAS
SEBELAS MARET , 2010.

- [10] “Introduction of Spring Framework,” Spring, [Online]. Available: <http://docs.spring.io/spring-framework/docs/3.0.x/reference/overview.html>. [Diakses 6 April 2015].
- [11] Oracle, “Oracle Java Documentation,” [Online]. Available: <https://docs.oracle.com/javase/tutorial/jdbc/overview/index.html>. [Diakses 5 Mei 2015].
- [12] “Tutorials Point,” [Online]. Available: http://www.tutorialspoint.com/spring/spring_transaction_management.htm. [Diakses 5 Mei 2015].
- [13] “The Java EE 5 Tutorial,” 5 Mei 2015. [Online]. Available: <https://docs.oracle.com/javaee/5/tutorial/doc/bnafe.html>.
- [14] “Tutorials Point,” [Online]. Available: http://www.tutorialspoint.com/spring/spring_bean_definition.htm. [Diakses 6 Mei 2015].
- [15] PostgreSQL, “PostgreSQL,” [Online]. Available: <http://www.postgresql.org/docs/9.3/static/intro-what-is.html>. [Diakses 26 Mei 2015].
- [16] H. P. Breivold, I. Crnkovic dan P. Eriksson, “Evaluating Software Evolvability,” *Proceedings of the 7th Conference on Software Engineering Research and Practice in Sweden*, vol. 7, pp. 96-103, 2007.

LAMPIRAN A. KODE SUMBER

1.	@Entity
2.	@Table(name="katalog_alihjenjang")
3.	public class KatalogAlihjenjang {
4.	
5.	@Id
6.	@Column(name="id_katalog_alihjenjang"
7.)
8.	@GeneratedValue(generator = "uuid")
9.	@GenericGenerator(name = "uuid",
	strategy = "uuid2")
10.	@org.hibernate.annotations.Type(type=
11.	"pg-uuid")
12.	private UUID idKatalogAlihjenjang;
13.	
	@NotEmpty(message="Nama Katalog tidak
	boleh kosong")
14.	@Column(name="nm_katalog")
15.	private String nmKatalog;
16.	
17.	@Column(name="catatan")
18.	private String catatan;
19.	
20.	@NotNull(message="Status hapus tidak
	boleh kosong")
21.	@Column(name="a_katalog_alihjenjang_t
	erhapus")
22.	private boolean aTerhapus;
23.	
24.	public UUID getIdKatalogAlihjenjang()
25.	{
26.	return idKatalogAlihjenjang;
27.	}
28.	
29.	public void
	setIdKatalogAlihjenjang(UUID
	idKatalogAlihjenjang) {

30.	this.idKatalogAlihjenjang =
31.	idKatalogAlihjenjang;
32.	}
33.	public String getNmKatalog() {
34.	return nmKatalog;
35.	}
36.	
37.	public void setNmKatalog(String
38.	nmKatalog) {
39.	this.nmKatalog = nmKatalog;
40.	}
41.	public String getCatatan() {
42.	return catatan;
43.	}
44.	
45.	public void setCatatan(String
46.	catatan) {
47.	this.catatan = catatan;
48.	}
49.	public boolean isaTerhapus() {
50.	return aTerhapus;
51.	}
52.	
53.	public void setaTerhapus(boolean
54.	aTerhapus) {
55.	this.aTerhapus = aTerhapus;
56.	}

Kode Sumber A.1 Kelas entitas katalog alih jenjang

1.	@Repository
2.	public class MKAlihjenjangRepositoryImpl
3.	implements MKAlihjenjangRepository{
4.	@Autowired
5.	

6.	private SessionFactory
7.	sessionFactory;
8.	@Override
9.	public List<MKAlihjenjang> get(String
10.	where, String order, int limit, int offset)
11.	{
12.	String dbWhere = "";
13.	String dbOrder = "";
14.	if(where != "") dbWhere = "
15.	WHERE "+where;
16.	if(order != "") dbOrder = "
17.	ORDER BY "+order;
18.	Query query =
19.	sessionFactory.getCurrentSession().createQu
20.	ery("select mkAlihjenjang from
21.	MKAlihjenjang mkAlihjenjang join "
22.	+
23.	"mkAlihjenjang.katalogAlihjenjang katalog"+
24.	dbWhere+dbOrder);
25.	if(limit != -1 && limit>0) {
26.	query.setFirstResult(offset);
27.	if(offset < 0) offset =
28.	0;
29.	query.setMaxResults(limit);
30.	}
31.	return query.list();
32.	}
33.	@Override
34.	public MKAlihjenjang getId(UUID
35.	idMKAlihjenjang) {
36.	List<MKAlihjenjang> queryResult
37.	=

	sessionFactory.getCurrentSession().createQuery("select mkAlihjenjang from
29.	MKAlihjenjang mkAlihjenjang WHERE "
	+
	"mkAlihjenjang.idMKAlihjenjang =
30.	'"+idMKAlihjenjang.toString()+"').list();
31.	if(queryResult.size()==0)
32.	return null;
33.	else
34.	{
35.	return
36.	queryResult.get(0);
37.	}
38.	}
39.	
40.	@Override
	public UUID insert(MKAlihjenjang
41.	mkAlihjenjang) {
	Session session =
42.	sessionFactory.openSession();
	Transaction tx =
43.	session.beginTransaction();
	UUID insertId=
44.	(UUID)session.save(mkAlihjenjang);
45.	tx.commit();
46.	session.flush();
47.	session.close();
48.	return insertId;
50.	}
51.	
52.	@Override
	public void update(MKAlihjenjang
53.	mkAlihjenjang) {
	Session session =
54.	sessionFactory.openSession();
	Transaction tx =
55.	session.beginTransaction();
56.	session.update(mkAlihjenjang);
57.	tx.commit();

58.	session.flush();
59.	session.close();
60.	}
61.	
62.	@Override
	public void delete(MKAlihjenjang
63.	mkAlihjenjang) {
	Session session =
64.	sessionFactory.openSession();
	Transaction tx =
65.	session.beginTransaction();
66.	session.delete(mkAlihjenjang);
67.	tx.commit();
68.	session.flush();
69.	session.close();
70.	}
71.	
72.	@Override
73.	public long count(String where) {
74.	String dbWhere = "";
	if(where != "") dbWhere = "
75.	WHERE "+where;
76.	Query query =
	sessionFactory.getCurrentSession().createQuery("select count(*) from MKAlihjenjang
	mkAlihjenjang join
	mkAlihjenjang.katalogAlihjenjang
80.	katalog"+dbWhere);
	Long count =
81.	(Long)query.uniqueResult();
82.	return count;
83.	}
	}

Kode Sumber A.2 Kelas *repository* MKAlihjenjangRepository

1.	@Service
2.	public class MKAlihjenjangServiceImpl
	implements MKAlihjenjangService{
3.	

```

4.         private String[] column =
           {"mkAlihjenjang.idMKAlihjenjang","katalog.n
           mKatalog","mkAlihjenjang.kodeMKAlihjenjang"
           ,"mkAlihjenjang.nmMKAlihjenjang",

5.           "mkAlihjenjang.jumlahSKS","mkAlihjenj
           ang.deskripsiMKAlihjenjang","mkAlihjenjang.
           aMKAlihjenjangTerhapus"};
6.         private Boolean[] searchable =
           {false,true,true,true,true,true,false};
7.
8.         @Autowired
9.         MKAlihjenjangRepository
           mkAlihjenjangRepository;
10.
11.        @Override
12.        public List<MKAlihjenjang> get() {
13.            return get("");
14.        }
15.
16.        @Override
17.        public List<MKAlihjenjang> get(String
           where) {
18.            return get(where,"");
19.        }
20.
21.        @Override
22.        public List<MKAlihjenjang> get(String
           where, String order) {
23.            return get(where,order,-1,-1);
24.        }
25.
26.        @Override
27.        public List<MKAlihjenjang> get(String
           where, String order, int limit,
28.            int offset) {
29.            return
           mkAlihjenjangRepository.get(where,order,lim
           it,offset);

```

```

30.         }
31.
32.         @Override
33.         public MKAlihjenjang getById(UUID
idMKAlihjenjang) {
34.             return
mkAlihjenjangRepository.getById(idMKAlihjen
jang);
35.         }
36.
37.
38.         @Override
39.         public String save(MKAlihjenjang
mkAlihjenjang) {
40.             if(mkAlihjenjang.getIdMKAlihjenjang()
!= null)
41.                 {
42.                     //update
43.
44.                     mkAlihjenjangRepository.update(mkAlih
jenjang);
45.                     return
mkAlihjenjang.getIdMKAlihjenjang().toString
();
46.                 }
47.             else
48.                 {
49.                     //insert
50.                     return
mkAlihjenjangRepository.insert(mkAlihjenjan
g).toString();
51.                 }
52.         }
53.
54.         @Override
55.         public String delete(UUID
idMKAlihjenjang) {
56.

```

```

57.         MKAlihjenjang mkAlihjenjang =
mkAlihjenjangRepository.getById(idMKAlihjen
58.         jang);
59.         if(mkAlihjenjang==null) return
null;
60.         else{
mkAlihjenjang.setaMKAlihjenjangTerhadap
61.         us(true);

62.         mkAlihjenjangRepository.update(mkAlih
63.         jenjang);
64.         return "Ok";
65.         }
66.     }
67.
@Override
68.     public Datatable getdatatable(String
sEcho, int iDisplayLength,
69.         int iDisplayStart, int
iSortCol_0, String sSortDir_0,
70.         String sSearch, String
filter) {
        DatatableExtractParams
parameter = new
DatatableExtractParams(sSearch,
71.         this.column, this.searchable, iSortCol_0,
sSortDir_0);
72.         Datatable mkLuarDatatable= new
73.         Datatable();
74.
mkLuarDatatable.setsEcho(sEcho);
75.         String dbFilter = "";
76.         if(filter != null &&
!filter.equals("")) dbFilter+=" AND
77.         "+filter;
        List<MKAlihjenjang> queryResult
=
get("(" + parameter.getWhere() + ") "+dbFilter,
78.

```

79.	parameter.getOrder(), iDisplayLength, iDisplayStart);
80.	List<String[]> aData = new ArrayList<String[]>();
81.	for (MKAlihjenjang mkAlihjenjang : queryResult) {
	String[]
	mkAlihjenjangString = new String[7];
82.	mkAlihjenjangString[0] = mkAlihjenjang.getIdMKAlihjenjang().toString ();
83.	mkAlihjenjangString[1] = String.valueOf(mkAlihjenjang.getKatalogAlih jenjang().getNmKatalog());
84.	mkAlihjenjangString[2] = String.valueOf(mkAlihjenjang.getKodeMKAlih jenjang());
85.	mkAlihjenjangString[3] = String.valueOf(mkAlihjenjang.getNmMKAlihjen jang());
86.	mkAlihjenjangString[4] = String.valueOf(mkAlihjenjang.getJumlahSKS());
87.	mkAlihjenjangString[5] = String.valueOf(mkAlihjenjang.getDeskripsiMK Alihjenjang());
88.	mkAlihjenjangString[6] =
89.	String.valueOf(mkAlihjenjang.isaMKAlihjenja
90.	ngTerhapus());
91.	
92.	aData.add(mkAlihjenjangString);
93.	}
94.	
	mkLuarDatatable.setAaData(aData);
95.	
96.	mkLuarDatatable.setiTotalRecords(mkAl ihjenjangRepository.count(""));
	mkLuarDatatable.setiTotalDisplayRecor
97.	

98.	<code>ds(mkAlihjenjangRepository.count("(" + paramete</code>
99.	<code>ter.getWhere() + ") " + dbFilter));</code>
	<code>return mkLuarDatatable;</code>
	<code>}</code>
	<code>}</code>

Kode Sumber A.3 Kelas *service* MKAlihjenjangService

1.	<code>public class AjaxResponse {</code>
2.	<code>private String status;</code>
3.	<code>private String message;</code>
4.	<code>private Object data;</code>
5.	<code>public AjaxResponse(String status,</code>
	<code>String message, Object data) {</code>
6.	<code> this.status = status;</code>
7.	<code> this.message = message;</code>
8.	<code> this.setData(data);</code>
9.	<code>}</code>
10.	<code>public AjaxResponse() {</code>
11.	<code> this.status = "ok";</code>
12.	<code> this.message = "";</code>
13.	<code> this.data = null;</code>
14.	<code>}</code>
15.	<code>public String getStatus() {</code>
16.	<code> return status;</code>
17.	<code>}</code>
18.	<code>public void setStatus(String</code>
19.	<code>status) {</code>
20.	<code> this.status = status;</code>
21.	<code>}</code>
22.	<code>public String getMessage() {</code>
23.	<code> return message;</code>
24.	<code>}</code>
25.	<code>public void setMessage(String</code>
	<code>message) {</code>
26.	<code> this.message = message;</code>
27.	<code>}</code>
28.	<code>public Object getData() {</code>
29.	<code> return data;</code>

30.	}
31.	public void setData(Object data) {
32.	this.data = data;
33.	}
34.	}

Kode Sumber A.4 Kelas AjaxResponse

1.	@RequestMapping(value = "saveekuivalen",
2.	method = RequestMethod.POST,
3.	produces="application/json")
4.	public @ResponseBody AjaxResponse
5.	saveekuivalen(HttpSession session,
6.	@RequestBody EkuivalenMKAlihjenjang[]
7.	ekuivalenMKAlihjenjang) {
8.	if(!isLogin(session)){ return
9.	null;}
10.	if(!hasMenu(session, "Kelola
11.	Katalog SatMan")) { return null;}
12.	KatalogAlihjenjang
13.	katalogAlihjenjang =
14.	katalogAlihjenjangService.getById(UUID.fromString(ekuivalenMKAlihjenjang[0].getIdKatalogAlihjenjang()));
15.	if(katalogAlihjenjang == null)
16.	return new AjaxResponse("error", "Data tidak
17.	ditemukan", null);
18.	Kurikulum kurikulum =
19.	kurikulumService.getById(UUID.fromString(ekuivalenMKAlihjenjang[0].getIdKurikulum()));
20.	if(kurikulum == null) return
21.	new AjaxResponse("error", "Data tidak
22.	ditemukan", null);
23.	//Delete semua relasi dan mk
24.	ekuivalensi
25.	List<RelasiMKAlihjenjang>
26.	listRelasiMKAlihjenjang =

	<pre> relasiMKAlihjenjangService.get("katalog.idK atalogAlihjenjang = '" 16. +katalogAlihjenjang.getIdKatalogAlihjenjang().toString()+"' AND kurikulum.idKurikulum = 17. '"+kurikulum.getIdKurikulum()+"'"); 18. for (RelasiMKAlihjenjang relasiMKAlihjenjang : listRelasiMKAlihjenjang) 19. { 20. List<EkuivalensiMKAlihjenjang> listEkuivalensiMKAlihjenjang = ekuivalensiMKAlihjenjangService.get("relasi MKAlihjenjang.idRelasiMKAlihjenjang = '" 21. +relasiMKAlihjenjang.getIdRelasiMKAlihjenjang().toString()+"'"); 22. 23. for (EkuivalensiMKAlihjenjang ekuivalensiMKAlihjenjang : listEkuivalensiMKAlihjenjang) { 24. ekuivalensiMKAlihjenjangService.delete(ekuivalensiMKAlihjenjang.getIdEkuivalensiMKAlihjenjang()); 25. } 26. 27. relasiMKAlihjenjangService.delete(relasiMKAlihjenjang.getIdRelasiMKAlihjenjang()); 28. } 29. 30. for(EkuivalenMKAlihjenjang mk : ekuivalenMKAlihjenjang) 31. { 32. int max = mk.getIdMKAlihjenjang().length; 33. if(max < mk.getIdMKSatMan().length) </pre>
--	--

```

34.                                     max =
mk.getIdMKSatMan().length;
35.                                     RelasiMKAlihjenjang
relasiMKAlihjenjang = new
RelasiMKAlihjenjang();
36.
37.         relasiMKAlihjenjang.setKatalog(katalo
gAlihjenjang);
38.
39.         relasiMKAlihjenjang.setKurikulum(kuri
kulum);
40.                                     String relasi =
mk.getRelasiMKAlihjenjang();
41.                                     relasi =
relasi.replaceAll("\\s+", "");
42.                                     String detailRelasi =
null;
43.
44.         mk.setRelasiMKAlihjenjang(relasi);
45.                                     if(relasi.length() > 2)
46.                                     {
47.                                     List<String>
listIDMK = new ArrayList<String>();
48.                                     for(int
i=0;i<mk.getIdMKAlihjenjang().length;i++)
49.                                     {
50.                                     String tmp =
mk.getIdMKAlihjenjang()[i];
51.
52.         listIDMK.add(mk.getIdMKAlihjenjang()[
i]);
53.                                     }
54.                                     detailRelasi =
generateDetailRelasi(relasi, listIDMK);
55.                                     }
56.
57.         relasiMKAlihjenjang.setRelasiMKAlihje
njang(mk.getRelasiMKAlihjenjang());
58.

```

59.	relasiMKAlihjenjang.setDetailRelasiMK
60.	Alihjenjang(detailRelasi);
61.	
	relasi =
62.	mk.getRelasiMKSatMan();
	relasi =
63.	relasi.replaceAll("\\s+", "");
64.	detailRelasi = null;
65.	
66.	mk.setRelasiMKSatMan(relasi);
67.	if(relasi.length() > 2)
68.	{
	List<String>
69.	listIDMK = new ArrayList<String>();
	for(int
70.	i=0;i<mk.getIdMKSatMan().length;i++)
71.	{
	String tmp =
72.	mk.getIdMKSatMan()[i];
73.	
74.	listIDMK.add(mk.getIdMKSatMan()[i]);
75.	}
	detailRelasi =
76.	generateDetailRelasi(relasi, listIDMK);
77.	}
78.	
	relasiMKAlihjenjang.setRelasiMK(mk.ge
79.	tRelasiMKSatMan());
80.	
	relasiMKAlihjenjang.setDetailRelasiMK
81.	(detailRelasi);
82.	
83.	
	relasiMKAlihjenjangService.save(relas
84.	iMKAlihjenjang);
	MKAlihjenjang
85.	mkAlihjenjang = new MKAlihjenjang();
86.	MK mkSatMan = new MK();

87.	for(int i=0;i<max;i++)
88.	{
89.	EkuivalensiMKAlihjenjang
	ekuivalensiMKAlihjenjang = new
90.	EkuivalensiMKAlihjenjang();
91.	
	if(i <
92.	mk.getIdMKAlihjenjang().length)
	mkAlihjenjang
	=
	mkAlihjenjangService.getById(UUID. <i>fromStrin</i>
93.	<i>g</i> (mk.getIdMKAlihjenjang()[i]));
	if(i <
94.	mk.getIdMKSatMan().length)
	mkSatMan =
	mkService.getById(UUID. <i>fromString</i> (mk.getIdM
95.	KSatMan()[i]));
96.	
	ekuivalensiMKAlihjenjang.setMk(mkSatM
97.	an);
98.	
	ekuivalensiMKAlihjenjang.setMkAlihjen
99.	jang(mkAlihjenjang);
100.	
	ekuivalensiMKAlihjenjang.setRelasiMKA
101.	lihjenjang(relasiMKAlihjenjang);
102.	
	ekuivalensiMKAlihjenjangService.save(
103.	ekuivalensiMKAlihjenjang);
104.	}
105.	}
	return new
	AjaxResponse("ok","Penyimpanan
106.	berhasil",null);
	}

**Kode Sumber A.5 Fungsi saveekuivalen pada kelas
KatalogSatManController**

```

1.      private boolean isoperator(char
cek_op)
2.      {
3.          if(cek_op=='^' || cek_op=='/')
4.          {
5.              return true;
6.          }
7.          else{
8.              return false;
9.          }
10.     }

11.     private int priority(char pri)
12.     {
13.         if(pri=='^'){
14.             return 2;
15.         }
16.         else if(pri=='/'){
17.             return 1;
18.         }
19.         else{
20.             return 0;
21.         }
22.     }
23.
24.     private String createpostfix(String
data)
25.     {
26.         Stack<Character> stack = new
Stack<Character>();
27.         Character cek=' ';
28.         data = data.replaceAll("\\s+" ,
29.         "");
30.         Character[] input = new
Character[data.length()];
31.         Character[] output = new
Character[data.length()];
32.
33.         for(int i=0;i<data.length();i++)

```

```

34.         {
35.             input[i] = new
Character(data.charAt(i));
36.         }
37.
38.         int k = 0;
39.         for(int i=0; i<data.length();
40. i++){
41.
42.         if(Character.isLetter(input[i])){
43.             output[k]=input[i];
44.             k++;
45.         }
46.         else
47.         if(isoperator(input[i]))
48.         {
49.             if(stack.size()==0){
50.
51.             stack.push(input[i]);
52.             }
53.             else
54.             {
55.                 if(priority(input[i]) <=
priority(stack.lastElement()) && cek!='(')
56.                 {
57.                     for(int
58. j=stack.size()-1; j>-1; j--)
59.                     {
60.                         if(!(stack.get(j) == '('))
61.                         {
62.
63.                         output[k]=stack.pop();
64.
65.                         k++;
66.                         }
67.                     }
68.                 stack.push(input[i]);

```

```

69.         }
70.         else
71.         {
72.
73.             stack.push(input[i]);
74.         }
75.     }
76. }
77. else if(input[i]=='(')
78. {
79.     stack.push(input[i]);
80.     cek=input[i];
81. }
82. else if(input[i]==')')
83. {
84.     cek=input[i];
85.
86. while(stack.lastElement() != '(')
87.     {
88.
89.         output[k]=stack.pop();
90.         k++;
91.     }
92.     if(stack.lastElement()
93. == '('){
94.
95.         output[k]=stack.pop();
96.     }
97. }
98.
99.         for(int i=stack.size()-1; i>=0;
100. i--){
101.
102.         if(!(stack.lastElement()=='(')){
103.             output[k]=stack.pop();
104.             k++;
105.         }
106.     }

```


107.	
108.	String hasil = "";
109.	for(int i=0;i<output.length;i++)
110.	{
111.	if(output[i] != null)
112.	{
113.	hasil += ""+output[i];
114.	}
115.	}
	return hasil;
	}

Kode Sumber A.6 Fungsi createpostfix pada CalonPDController

[Halaman ini sengaja dikosongkan]

LAMPIRAN B. TABEL

Tabel B.1 Rincian kasus penggunaan kelola katalog alih jenjang

komponen	deskripsi
Nama	Mengelola katalog alih jenjang
Nomor	UC1
Deskripsi	Kasus penggunaan ini digunakan untuk mengatur katalog alih jenjang. Pengelolaan katalog alih jenjang meliputi nama katalog, dan catatan.
Tipe	Fungsional
Pemicu	Aktor memilih menu manajemen katalog alih jenjang.
Aktor	Administrator
Kondisi Awal	-
Kondisi Akhir	Katalog alih jenjang terbuat.
Alur Normal	<ol style="list-style-type: none"> 1. Aktor memilih menu kelola katalog alih jenjang. 2. Sistem menampilkan daftar semua katalog alih jenjang . 3. Aktor memilih menambah katalog alih jenjang. 4. Sistem menampilkan formulir katalog alih jenjang yang terdiri atas: <ul style="list-style-type: none"> - nama katalog, - catatan. 5. Aktor mengisi formulir katalog alih jenjang. 6. Sistem menyimpan katalog alih jenjang baru.
komponen	deskripsi
Alur Alternatif	<ol style="list-style-type: none"> 3.1. Aktor memilih menyunting katalog alih jenjang <ol style="list-style-type: none"> 1. Sistem menampilkan formulir katalog alih jenjang yang terdiri atas: <ul style="list-style-type: none"> - nama katalog, - catatan. 2. Aktor menyunting formulir katalog alih jenjang. 3. Sistem menyimpan perubahan katalog alih jenjang. 3.2. Aktor memilih menghapus katalog alih jenjang <ol style="list-style-type: none"> 1. Sistem menghapus katalog alih jenjang 1.1. Aktor membatalkan penambahan katalog alih jenjang baru.

	<ol style="list-style-type: none"> 1. Kasus penggunaan berakhir. 3.1.2.1. Aktor membatalkan penyuntingan katalog alih jenjang 1. Kasus penggunaan berakhir
--	---

Tabel B.2 Rincian kasus penggunaan kelola matakuliah alih jenjang

komponen	deskripsi
Nama	Mengelola matakuliah alih jenjang
Nomor	UC2
Deskripsi	Kasus penggunaan ini digunakan untuk mengatur matakuliah alih jenjang. Pengelolaan matakuliah alih jenjang meliputi kode matakuliah, nama matakuliah, jumlah sks, dan deskripsi matakuliah.
Tipe	Fungsional
Pemicu	Aktor memilih menu kelola matakuliah alih jenjang.
Aktor	Administrator
Kondisi Awal	-
Kondisi Akhir	Matakuliah alih jenjang terbuat.
Alur Normal	<ol style="list-style-type: none"> 1. Aktor memilih menu kelola matakuliah alih jenjang. 2. Sistem menampilkan daftar semua matakuliah alih jenjang. 3. Aktor memilih menambah matakuliah alih jenjang. 4. Sistem menampilkan formulir matakuliah alih jenjang yang terdiri atas: <ul style="list-style-type: none"> - kode matakuliah, - nama matakuliah, - jumlah sks, - deskripsi. 5. Aktor mengisi formulir matakuliah alih jenjang. 6. Sistem menyimpan matakuliah alih jenjang baru.
Alur Alternatif	<ol style="list-style-type: none"> 3.3. Aktor memilih menyunting matakuliah alih jenjang <ol style="list-style-type: none"> 1. Sistem menampilkan formulir matakuliah alih jenjang yang terdiri atas: <ul style="list-style-type: none"> - kode matakuliah, - nama matakuliah,

	<ul style="list-style-type: none"> - jumlah sks, - deskripsi. <ol style="list-style-type: none"> 2. Aktor menyunting formulir matakuliah alih jenjang. 3. Sistem menyimpan perubahan matakuliah alih jenjang. 3.4. Aktor memilih menghapus matakuliah alih jenjang <ol style="list-style-type: none"> 1. Sistem menghapus matakuliah alih jenjang 1.2. Aktor membatalkan penambahan matakuliah alih jenjang baru. <ol style="list-style-type: none"> 1. Kasus penggunaan berakhir. 3.1.2.2. Aktor membatalkan penyuntingan matakuliah alih jenjang <ol style="list-style-type: none"> 2. Kasus penggunaan berakhir
--	---

Tabel B.3 Rincian kasus penggunaan membagikan katalog ke satuan manajemen

komponen	deskripsi
Nama	Membagikan katalog ke satuan manajemen
Nomor	UC3
Deskripsi	Kasus penggunaan ini digunakan memberikan hak akses kepada satuan manajemen ke katalog agar dapat mengelola pemetaan matakuliah.
Tipe	Fungsional
Pemicu	Aktor memilih menu kelola katalog satman.
Aktor	Administrator
Kondisi Awal	Katalog tidak dapat diakses dari satuan manajemen
Kondisi Akhir	Katalog dapat diakses dari satuan manajemen
Alur Normal	<ol style="list-style-type: none"> 1. Aktor memilih menu kelola katalog satman. 2. Sistem menampilkan daftar akses satuan manajemen. 3. Aktor memilih menambah akses satuan manajemen. 4. Sistem menampilkan formulir akses satuan manajemen yang terdiri atas: <ul style="list-style-type: none"> - katalog, - satuan manajemen. 5. Aktor mengisi formulir akses katalog.

	6. Sistem menyimpan akses katalog..
Alur Alternatif	<p>3.5. Aktor memilih menghapus akses satuan manajemen.</p> <ol style="list-style-type: none"> 1. Sistem menghapus akses satuan manajemen. 1.3. Aktor membatalkan penambahan akses satuan manajemen. 1. Kasus penggunaan berakhir.

Tabel B.4 Rincian kasus penggunaan mengelola peserta didik alih jenjang

komponen	deskripsi
Nama	Mengelola peserta didik alih jenjang
Nomor	UC5
Deskripsi	Kasus penggunaan ini digunakan untuk mengatur peserta didik alih jenjang. Pengelolaan peserta didik alih jenjang meliputi nama, perguruan tinggi asal, alamat, tempat dan tanggal lahir.
Tipe	Fungsional
Pemicu	Aktor memilih menu manajemen mahasiswa alih jenjang.
Aktor	Administrator
Kondisi Awal	-
Kondisi Akhir	Peserta didik alih jenjang terbuat.
Alur Normal	<ol style="list-style-type: none"> 1. Aktor memilih menu manajemen mahasiswa alih jenjang. 2. Sistem menampilkan daftar peserta didik alih jenjang yang dimiliki satuan manajemen. 3. Aktor memilih menambah peserta didik alih jenjang. 4. Sistem menampilkan formulir peserta didik alih jenjang yang terdiri atas: <ul style="list-style-type: none"> - nama, - perguruan tinggi asal, - alamat, - tempat dan tanggal lahir. 5. Aktor mengisi formulir matakuliah alih jenjang. 6. Sistem menyimpan matakuliah alih jenjang baru.

Alur Alternatif	<p>3.1 Aktor memilih menyunting peserta didik alih jenjang</p> <ol style="list-style-type: none"> 1. Sistem menampilkan formulir peserta didik alih jenjang yang terdiri atas: <ul style="list-style-type: none"> - nama, - perguruan tinggi asal, - alamat, - tempat dan tanggal lahir. 2. Aktor menyunting formulir peserta didik alih jenjang. 3. Sistem menyimpan perubahan peserta didik alih jenjang. <p>3.2 Aktor memilih menghapus peserta didik alih jenjang</p> <ol style="list-style-type: none"> 8.2 Sistem menghapus peserta didik alih jenjang 1.4. Aktor membatalkan penambahan peserta didik alih jenjang baru. <ol style="list-style-type: none"> 1. Kasus penggunaan berakhir. 3.1.2.3. Aktor membatalkan penyuntingan. <ol style="list-style-type: none"> .1 Kasus penggunaan berakhir
------------------------	--

Tabel B.5 Rincian kasus penggunaan mengelola ekuivalensi peserta didik alih jenjang

komponen	deskripsi
Nama	Mengelola ekuivalensi matakuliah tempuh
Nomor	UC6
Deskripsi	Kasus penggunaan ini digunakan untuk melakukan proses ekuivalensi peserta didik alih jenjang. Proses ekuivalensi meliputi penambahan matakuliah yang sudah ditempuh dan proses ekuivalensi itu sendiri.
Tipe	Fungsional
Pemicu	Aktor memilih menu kelola ekuivalensi.
Aktor	Tim Ekuivalensi
Kondisi Awal	-
Kondisi Akhir	Matakuliah yang diakui terupdate
Alur Normal	<ol style="list-style-type: none"> 1. Aktor memilih menu manajemen mahasiswa alih jenjang.

	<ol style="list-style-type: none"> 2. Sistem menampilkan daftar peserta didik alih jenjang yang dimiliki satuan manajemen. 3. Aktor memilih kelola ekuivalensi. 4. Sistem menampilkan matakuliah tempuh dan matakuliah terakui. 5. Aktor menambahkan matakuliah tempuh. 6. Sistem menambah matakuliah tempuh. 7. Aktor memilih cek penyetaraan. 8. Sistem memperbarui matakuliah terakui.
komponen	deskripsi
Alur Alternatif	-

Tabel B.6 Rincian kasus penggunaan verifikasi matakuliah wajib

komponen	deskripsi
Nama	Verifikasi matakuliah wajib
Nomor	UC7
Deskripsi	Kasus penggunaan ini digunakan untuk melakukan proses verifikasi matakuliah wajib untuk peserta didik alih jenjang.
Tipe	Fungsional
Pemicu	Aktor memilih menu ekuivalensi alih jenjang.
Aktor	Tim ekuivalensi
Kondisi Awal	-
Kondisi Akhir	Menghasilkan matakuliah wajib untuk peserta didik alih jenjang
Alur Normal	<ol style="list-style-type: none"> 1. Aktor memilih menu ekuivalensi alih jenjang. 2. Sistem menampilkan daftar peserta didik alih jenjang. 3. Aktor memilih kelola matakuliah wajib. 4. Sistem menampilkan daftar kurikulum. 5. Aktor memilih kurikulum. 6. Sistem menampilkan daftar matakuliah. 7. Aktor mewajibkan matakuliah. 8. Sistem mewajibkan matakuliah. 9. Aktor menyimpan matakuliah wajib tempuh. 10. Sistem menyimpan matakuliah wajib tempuh.

komponen	Deskripsi
Alur Alternatif	<p>7.1 Aktor membebaskan matakuliah.</p> <ol style="list-style-type: none"> 1 Sistem membebaskan matakuliah. 2 Aktor menyimpan matakuliah wajib tempuh. 3 Sistem menyimpan matakuliah wajib tempuh 4 Kasus penggunaan berakhir. <p>7.2 Aktor membatalkan penyimpanan.</p> <ol style="list-style-type: none"> 1. Kasus penggunaan berakhir.

Tabel B.7 Rincian kasus penggunaan mencetak laporan Ekuivalensi alih jenjang

komponen	deskripsi
Nama	Mencetak laporan ekuivalensi alih jenjang
Nomor	UC8
Deskripsi	Kasus penggunaan ini digunakan untuk melakukan proses cetak laporan ekuivalensi.
Tipe	Fungsional
Pemicu	Aktor memilih menu laporan alih jenjang.
Aktor	Administrator
Kondisi Awal	-
Kondisi Akhir	Menghasilkan laporan versi cetak
Alur Normal	<ol style="list-style-type: none"> 1. Aktor memilih menu laporan alih jenjang. 2. Sistem menampilkan halaman laporan alih jenjang. 3. Aktor memilih peserta didik alih jenjang. 4. Sistem menampilkan daftar peserta didik alih jenjang. 5. Aktor mengklik cetak. 6. Sistem menampilkan laporan ekuivalensi versi cetak.
komponen	deskripsi
Alur Alternatif	-

Tabel B.8 Rincian kasus penggunaan mengelola aturan ekuivalensi

komponen	deskripsi
Nama	Mengelola aturan ekuivalensi
Nomor	UC9
Deskripsi	Kasus penggunaan ini digunakan untuk melakukan pemetaan matakuliah alih jenjang terhadap matakuliah satuan manajemen.
Tipe	Fungsional
Pemicu	Aktor memilih menu ekuivalensi kurikulum.
Aktor	Tim Ekuivalensi
Kondisi Awal	-
Kondisi Akhir	Matakuliah terpetakan.
Alur Normal	<ol style="list-style-type: none"> 1. Aktor memilih menu ekuivalensi kurikulum. 2. Sistem menampilkan riwayat ekuivalensi kurikulum. 3. Aktor memilih kelola ekuivalensi. 4. Sistem menampilkan halaman pengelolaan ekuivalensi. 5. Aktor memilih kelola ekuivalensi. 6. Sistem menampilkan halaman pengelolaan ekuivalensi. 7. Aktor menambah aturan. 8. Aktor menyimpan aturan. 9. Sistem menambah aturan. 10. Aktor menyimpan daftar aturan. 11. Sistem menyimpan daftar aturan.
komponen	deskripsi
Alur Alternatif	<ol style="list-style-type: none"> 7.1 Aktor menghapus riwayat aturan. <ol style="list-style-type: none"> 1. Aktor menyimpan daftar aturan. 2. Sistem menyimpan daftar aturan. 7.2 Aktor membatalkan penyimpanan daftar aturan. <ol style="list-style-type: none"> 1. Kasus penggunaan berakhir. 8.1 Aktor membatalkan penambahan aturan.

Tabel B.9 Rincian kasus penggunaan verifikasi ekuivalensi

komponen	deskripsi
Nama	Verifikasi Ekuivalensi
Nomor	UC10
Deskripsi	Kasus penggunaan ini digunakan untuk melakukan proses verifikasi ekuivalensi.
Tipe	Fungsional
Pemicu	Aktor memilih menu persetujuan ekuivalensi.
Aktor	Tim ekuivalensi
Kondisi Awal	-
Kondisi Akhir	Matakuliah yang ditempuh peserta didik terekuivalensi.
Alur Normal	<ol style="list-style-type: none"> 1. Aktor memilih menu persetujuan ekuivalensi. 2. Sistem menampilkan daftar mahasiswa yang ditangani. 3. Aktor menambahkan mahasiswa. 4. Sistem menampilkan hasil ekuivalensi. 5. Aktor menyimpan permanen hasil ekuivalensi. 6. Sistem menyimpan hasil ekuivalensi.
komponen	deskripsi
Alur Alternatif	<ol style="list-style-type: none"> 3.1. Aktor menyunting riwayat ekuivalensi. <ol style="list-style-type: none"> 1. Sistem menampilkan hasil ekuivalensi 2. Aktor menyimpan hasil ekuivalensi 3. Kasus penggunaan berakhir 3.2. Aktor membatalkan persetujuan ekuivalensi <ol style="list-style-type: none"> 1. Kasus penggunaan berakhir 5.1. Aktor membatalkan penyimpanan ekuivalensi <ol style="list-style-type: none"> 1. Kasus penggunaan berakhir

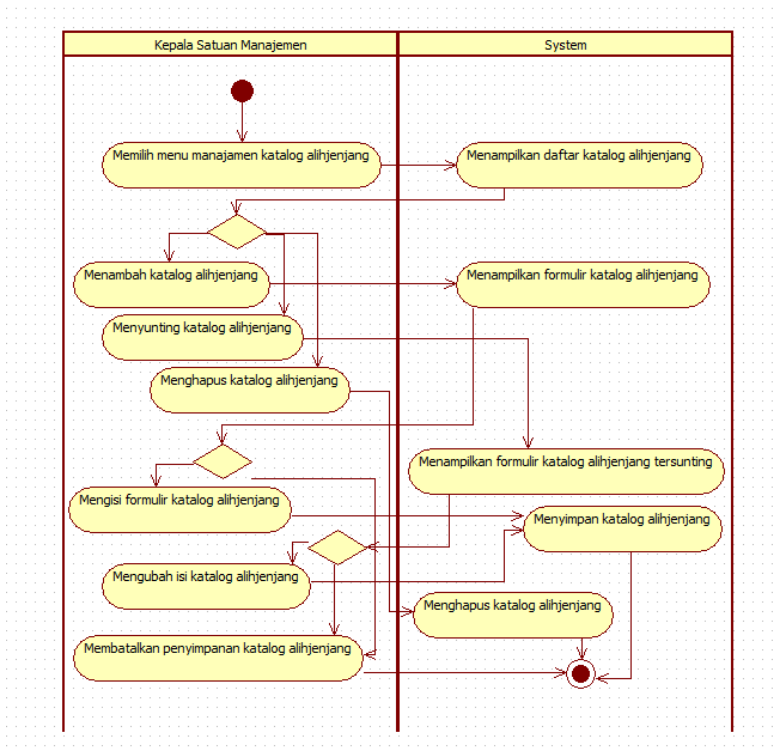
Tabel B.10 Rincian kasus penggunaan mencetak laporan ekuivalensi

komponen	deskripsi
Nama	Mencetak laporan ekuivalensi
Nomor	UC11

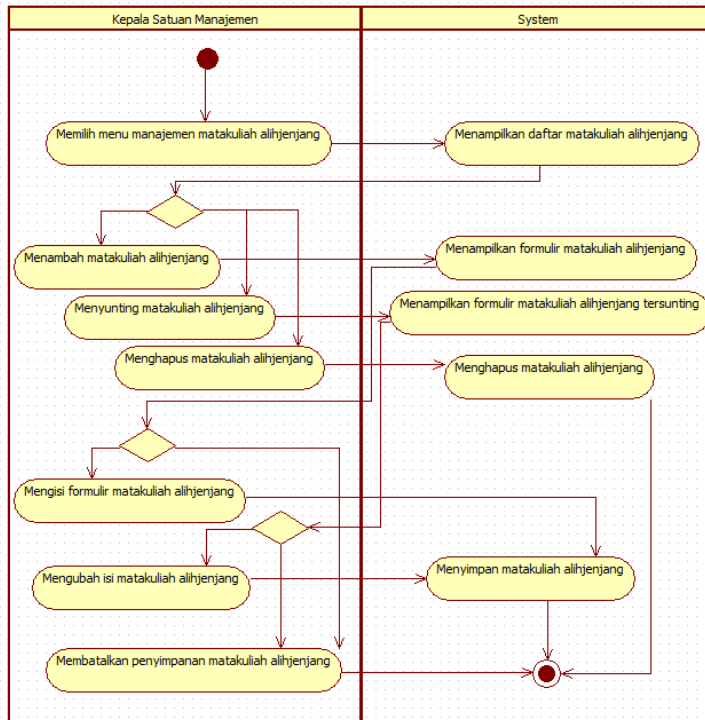
Deskripsi	Kasus penggunaan ini digunakan untuk melakukan proses cetak laporan ekuivalensi.
Tipe	Fungsional
Pemicu	Aktor memilih menu laporan ekuivalensi.
Aktor	Peserta didik
Kondisi Awal	-
Kondisi Akhir	Menghasilkan laporan versi cetak
Alur Normal	<ol style="list-style-type: none"> 1. Aktor memilih menu laporan ekuivalensi. 2. Sistem menampilkan halaman laporan ekuivalensi. 3. Aktor memilih kurikulum. 4. Sistem menampilkan daftar kurikulum. 5. Aktor memilih peserta didik. 6. Sistem menampilkan daftar peserta didik. 7. Sistem menampilkan hasil ekuivalensi. 8. Aktor mengklik menu cetak. 9. Sistem menampilkan laporan ekuivalensi versi cetak.
komponen	deskripsi
Alur Alternatif	-

[Halaman ini sengaja dikosongkan]

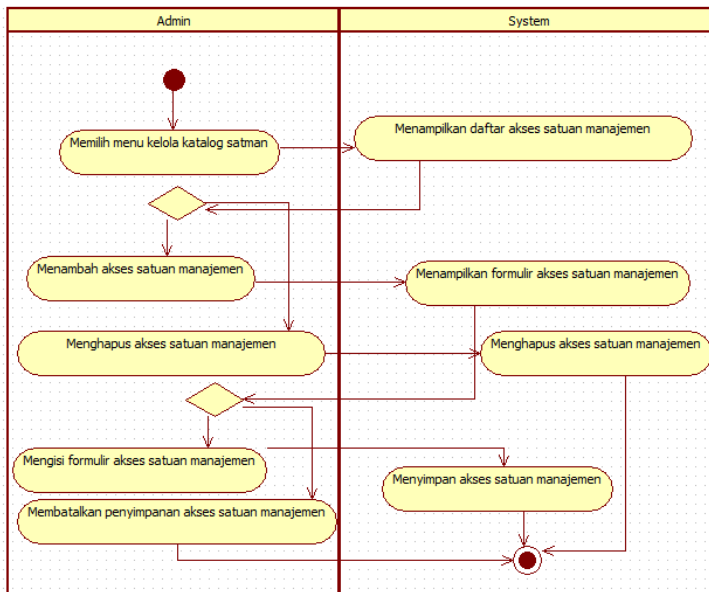
LAMPIRAN C. GAMBAR



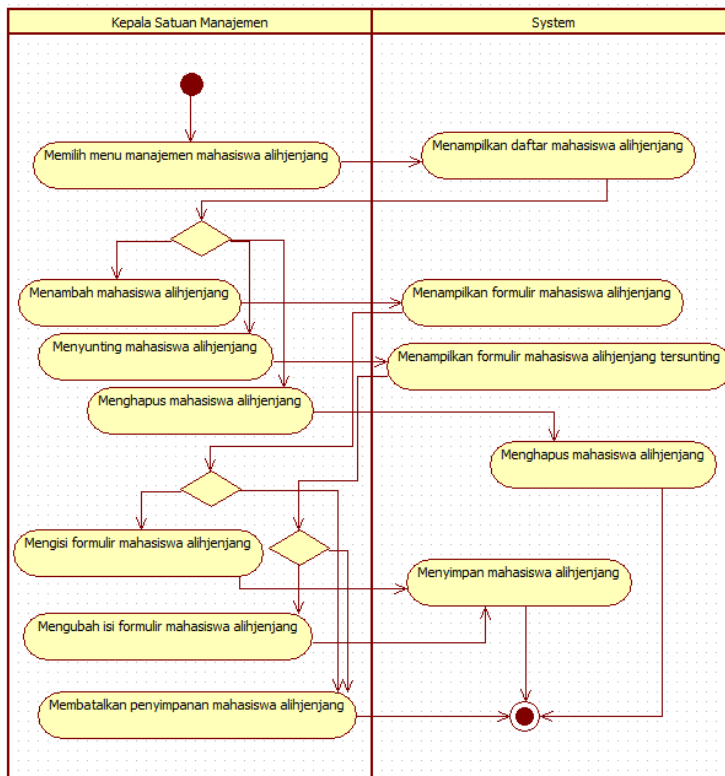
Gambar C.1 Kasus penggunaan mengelola katalog alih jenjang



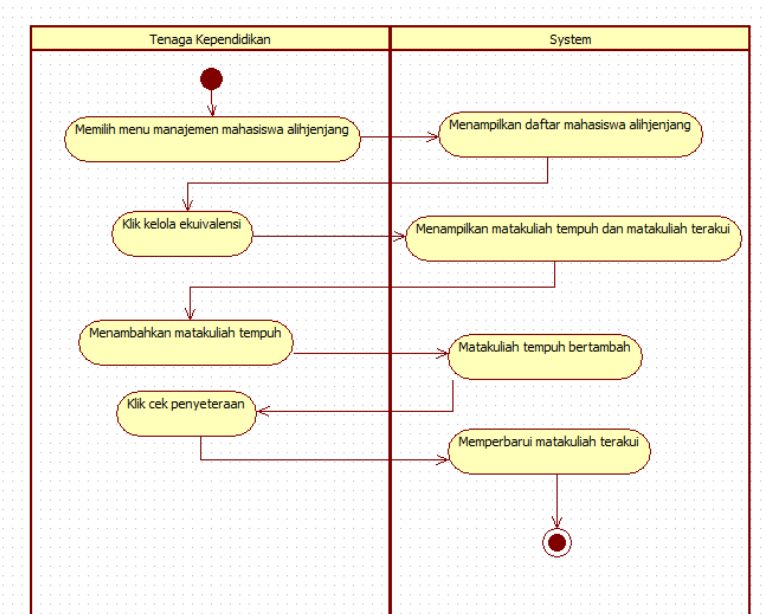
Gambar C.2 Kasus penggunaan mengelola matakuliah alih jenjang



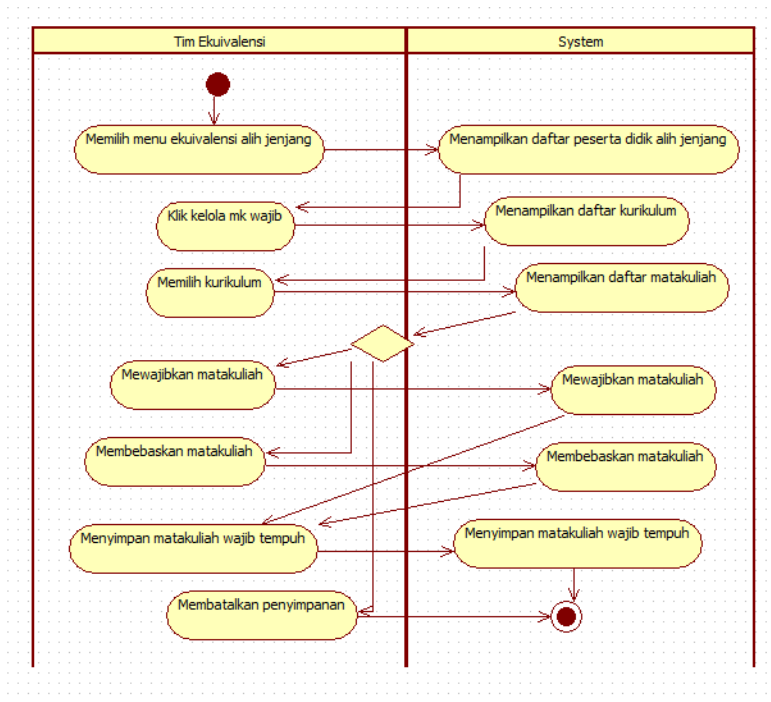
Gambar C.3 Kasus penggunaan membagikan katalog ke satuan manajemen



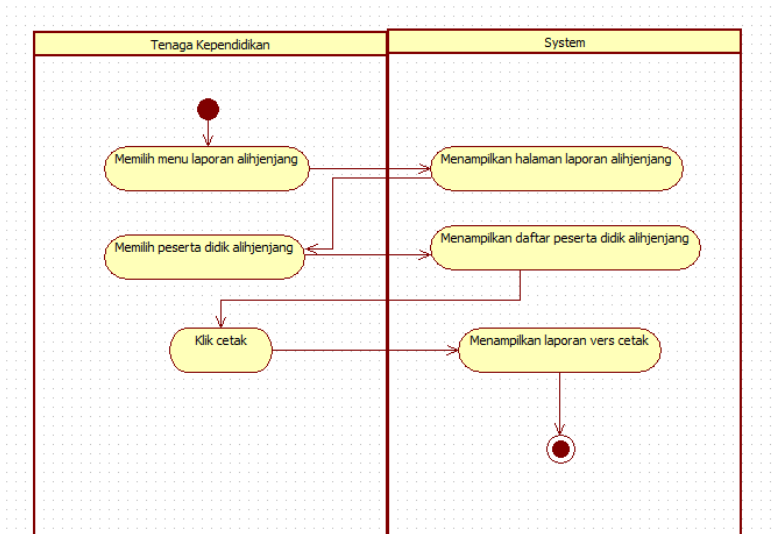
Gambar C.4 Kasus penggunaan mengelola peserta didik alih jenjang



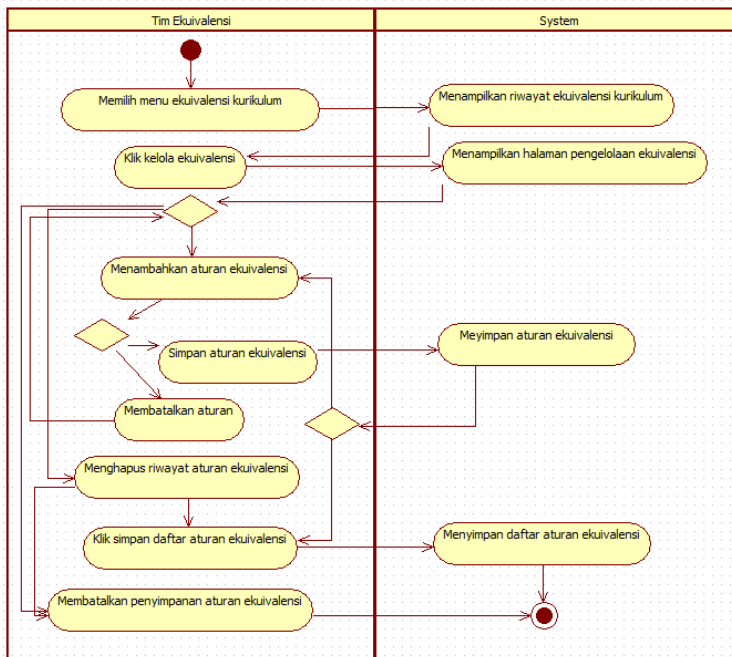
Gambar C.5 Kasus penggunaan mengelola ekuivalensi peserta didik alih jenjang



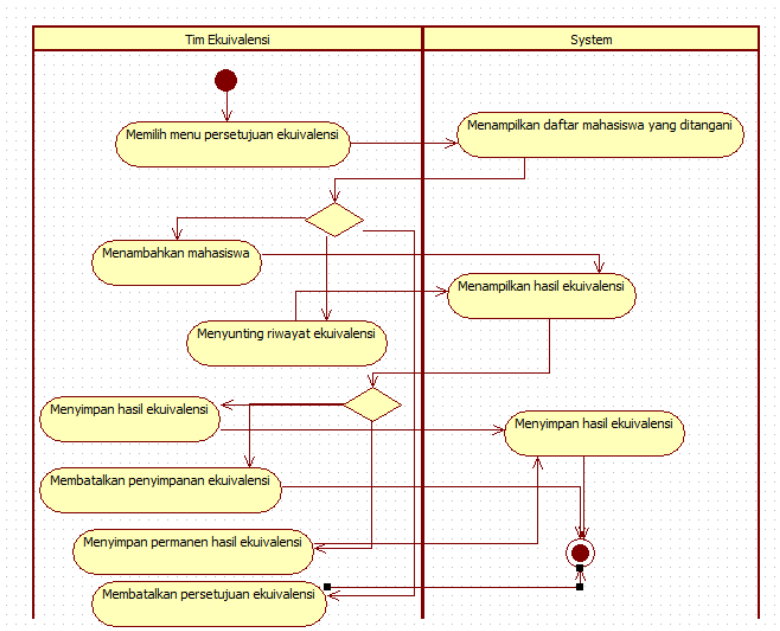
Gambar C.6 Kasus penggunaan verifikasi matakuliah wajib



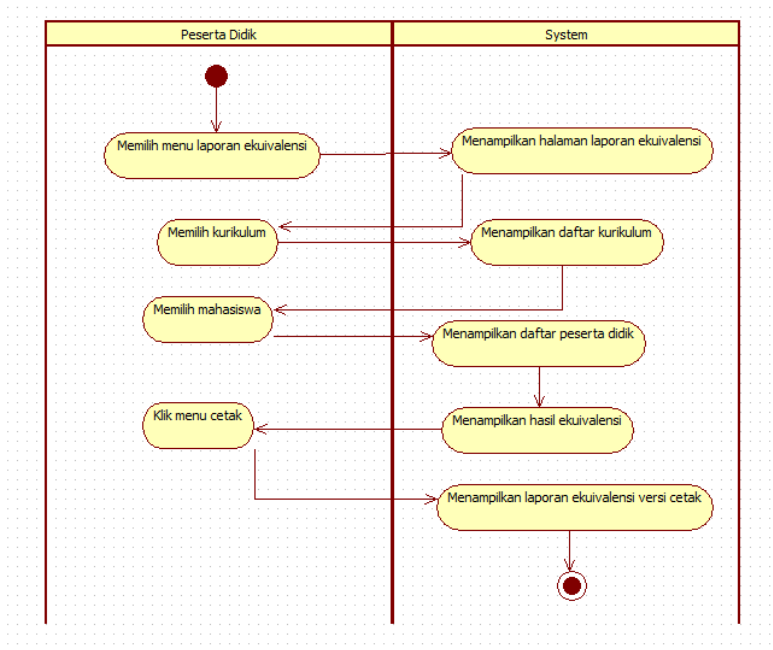
Gambar C.7 Kasus penggunaan mencetak laporan ekuivalensi alih jenjang



Gambar C.8 Kasus penggunaan mengelola aturan ekuivalensi



Gambar C.9 Kasus penggunaan verifikasi ekuivalensi



Gambar C.10 Kasus penggunaan mencetak laporan ekuivalensi

LEMBAR FEEDBACK

Nama	RA
Peran	

Terima kasih atas partisipasi anda dalam pengujian sistem informasi akademik pada modul penilaian. Untuk tahap akhir pengujian ini, kami meminta kesediaan anda untuk memberikan *feedback* kepada kami tentang pengalaman anda menggunakan aplikasi kami.

PERTANYAAN

1. Apakah desain antar muka dalam aplikasi ini mudah untuk dimengerti dan membantu anda untuk menggunakan aplikasi ini? Jika tidak, tolong sebutkan di bagian menu apa anda mengalami kesulitan beserta alasan atau saran anda.

Mudah dimengerti

2. Apakah dalam menggunakan aplikasi ini anda memiliki keluhan dalam hal kecepatan aplikasi? Jika ya, tolong sebutkan di bagian menu apa anda memiliki keluhan.

Sudah cukup

3. Apakah dalam proses bisnis yang anda jalankan pada skenario-skenario sudah sesuai dengan proses bisnis di tempat anda sedang berada? Jika tidak, tolong sebutkan di bagian menu apa terdapat proses bisnis yang berbeda dengan tempat anda.

Sudah sesuai

Gambar C.11 Lembar *feedback*

LEMBAR FEEDBACK	
Nama	Dini Adhi N
Peran	Dosen Ekuivalensi

Terima kasih atas partisipasi anda dalam pengujian sistem informasi akademik pada modul penilaian. Untuk tahap akhir pengujian ini, kami meminta kesediaan anda untuk memberikan *feedback* kepada kami tentang pengalaman anda menggunakan aplikasi kami.

PERTANYAAN

1. Apakah desain antar muka dalam aplikasi ini mudah untuk dimengerti dan membantu anda untuk menggunakan aplikasi ini? Jika tidak, tolong sebutkan di bagian menu apa anda mengalami kesulitan beserta alasan atau saran anda.

Cukup dimengerti, namun ada beberapa proses yang ~~kurang~~ perlu dijelaskan dengan contoh real

Contoh: - Status katalog aktif

- Relasi
2. Apakah dalam menggunakan aplikasi ini anda memiliki keluhan dalam hal kecepatan aplikasi? Jika ya, tolong sebutkan di bagian menu apa anda memiliki keluhan.

Tidak ada
3. Apakah dalam proses bisnis yang anda jalankan pada skenario-skenario sudah sesuai dengan proses bisnis di tempat anda sedang berada? Jika tidak, tolong sebutkan di bagian menu apa terdapat proses bisnis yang berbeda dengan tempat anda.

Sebagian besar sesuai.

Ada beberapa yang belum, seperti:

 - Info sts yang diambil, sts tempuh
 - Menampilkan MK ~~baru~~ di kurikulum lama setara dengan MK di kurikulum baru

Gambar C.12 Lembar *feedback*

SKENARIO UJI COBA MODUL EKUIVALENSI

Nama	RA
Peran	

Pada lembar skenario ini, akan ditampilkan daftar kebutuhan fungsional yang harus dipenuhi oleh sistem informasi akademik pada modul penilaian. Dalam setiap kebutuhan fungsional pada daftar tersebut akan terdapat beberapa skenario yang akan dipandu oleh pengembang sistem. Tugas anda di sini adalah untuk mencoba menjalankan skenario tersebut dan menentukan apakah kebutuhan fungsional tersebut dapat terpenuhi atau tidak. Silahkan jalankan skenario berdasarkan peran yang anda miliki saja.

KEPALA

No	Kebutuhan Fungsional	Kondisi	Terpenuhi
1.1	Sistem harus memiliki kemampuan untuk mengelola katalog alihjenjang	-	✓
1.2	Sistem harus memiliki kemampuan untuk mengelola matakuliah yang dimiliki katalog alihjenjang tersebut	Diakses melalui menu manajemen katalog alihjenjang	✓
		Diakses melalui menu manajemen matakuliah alihjenjang	✓
1.3	Sistem memiliki kemampuan untuk mengelola ekuivalensi atau pemetaan matakuliah di katalog terhadap matakuliah yang dimiliki jurusan	Tanpa relasi	✓
		Menggunakan operator AND(*)	✓
		Menggunakan operator OR(/)	✓

Gambar C.13 Lembar kuisisioner

		Menggunakan kombinasi AND(*) dan OR(/)	✓
1.3.1	Sistem menyediakan penjelasan mengenai apa itu relasi	-	✓
1.4	Sistem harus memiliki kemampuan untuk mengelola mahasiswa alihjenjang	Menggunakan proses penyetaraan matakuliah	✓
		Tanpa proses penyetaraan matakuliah	✓
1.4.1	Sistem dapat memasukkan matakuliah yang sudah ditempuh oleh mahasiswa alihjenjang	-	✓
1.4.2	Sistem dapat melakukan penyetaraan matakuliah sesuai pemetaan yang dilakukan sebelumnya	-	✓
1.4.3	Sistem dapat mengelola matakuliah wajib bagi mahasiswa alihjenjang	Matakuliah yang sudah terakui otomatis menjadi bebas ambil	✓
1.4.4	Sistem menyediakan laporan matakuliah yang wajib diambil mahasiswa alihjenjang setelah melakukan simpan permanen	-	✓
1.4.5	Sistem dapat membatalkan simpan permanen sehingga data matakuliah yang wajib diambil masih bisa diubah	-	✓

TIM EKUIVALENSI

No	Kebutuhan Fungsional	Kondisi	Terpenuhi
1.1	Sistem memiliki kemampuan untuk mengelola ekuivalensi saat pergantian kurikulum		✓

Gambar C.14 Lembar kuisioner

SKENARIO UJI COBA MODUL EKVIVALENSI

Nama	Dini Adni N
Peran	Dosen Ekvivalensi

Pada lembar skenario ini, akan ditampilkan daftar kebutuhan fungsional yang harus dipenuhi oleh sistem informasi akademik pada modul penilaian. Dalam setiap kebutuhan fungsional pada daftar tersebut akan terdapat beberapa skenario yang akan dipandu oleh pengembang sistem. Tugas anda di sini adalah untuk mencoba menjalankan skenario tersebut dan menentukan apakah kebutuhan fungsional tersebut dapat terpenuhi atau tidak. Silahkan jalankan skenario berdasarkan peran yang anda miliki saja.

KEPALA

No	Kebutuhan Fungsional	Kondisi	Terpenuhi
1.1	Sistem harus memiliki kemampuan untuk mengelola katalog alihjenjang	-	✓
1.2	Sistem harus memiliki kemampuan untuk mengelola matakuliah yang dimiliki katalog alihjenjang tersebut	Diakses melalui menu manajemen katalog alihjenjang	✓
		Diakses melalui menu manajemen matakuliah alihjenjang	✓
1.3	Sistem memiliki kemampuan untuk mengelola ekvivalensi atau pemetaan matakuliah di katalog terhadap matakuliah yang dimiliki jurusan	Tanpa relasi	✓
		Menggunakan operator AND(^)	✓
		Menggunakan operator OR(/)	✓

Gambar C.15 Lembar kuisioner

		Menggunakan kombinasi AND(^) dan OR(/)	✓
1.3.1	Sistem menyediakan penjelasan mengenai apa itu relasi	-	✓
1.4	Sistem harus memiliki kemampuan untuk mengelola mahasiswa alihjenjang	Menggunakan proses penyetaraan matakuliah	✓
		Tanpa proses penyetaraan matakuliah	✓
1.4.1	Sistem dapat memasukkan matakuliah yang sudah ditempuh oleh mahasiswa alihjenjang	-	✓
1.4.2	Sistem dapat melakukan penyetaraan matakuliah sesuai pemetaan yang dilakukan sebelumnya	-	✓
1.4.3	Sistem dapat mengelola matakuliah wajib bagi mahasiswa alihjenjang	Matakuliah yang sudah terakumulasi otomatis menjadi bebas ambil	✓
1.4.4	Sistem menyediakan laporan matakuliah yang wajib diambil mahasiswa alihjenjang setelah melakukan simpan permanen	-	✓
1.4.5	Sistem dapat membatalkan simpan permanen sehingga data matakuliah yang wajib diambil masih bisa diubah	-	✓

TIM EKVIVALENSI

No	Kebutuhan Fungsional	Kondisi	Terpenuhi
1.1	Sistem memiliki kemampuan untuk mengelola ekuivalensi saat pergantian kurikulum		✓

Gambar C.16 Lembar kuisioner

2.1.1	Sistem memiliki kemampuan untuk memetakan matakuliah pada kurikulum yang lama dengan matakuliah pada kurikulum yang baru	Tanpa relasi	✓
		Menggunakan operator AND(^)	✓
		Menggunakan operator OR(/)	✓
		Menggunakan kombinasi AND(^) dan OR(/)	✓
2.2	Sistem dapat mengelola persetujuan ekuivalensi pada tiap-tiap mahasiswa	-	✓
2.2.1	Sistem dapat mengelola matakuliah yang wajib diambil pada kurikulum baru dan dapat menghapus krs lulus yang sudah pernah ditempuh	-	✓
2.2.2	Sistem menyediakan laporan matakuliah yang wajib diambil mahasiswa alihjenjang setelah melakukan simpan permanen	-	✓
2.2.3	Sistem dapat membatalkan simpan permanen sehingga data matakuliah yang wajib diambil masih bisa diubah	-	✓

MAHASISWA

No	Kebutuhan Fungsional	Kondisi	Terpenuhi
1	Membuka laporan ekuivalensi yang sudah dilakukan mahasiswa	-	✓

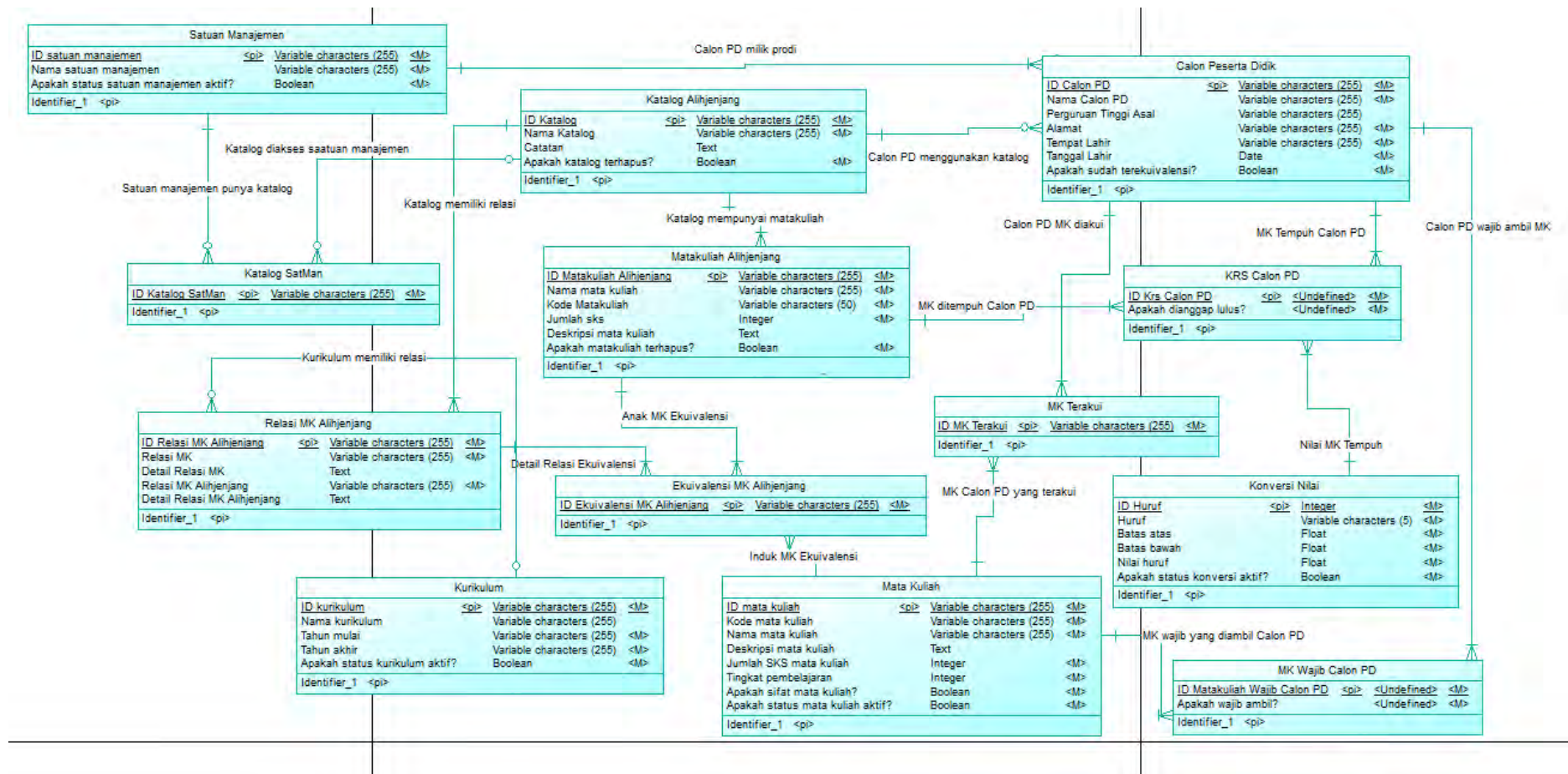
Gambar C.17 Lembar kuisisioner

2.1.1	Sistem memiliki kemampuan untuk memetakan matakuliah pada kurikulum yang lama dengan matakuliah pada kurikulum yang baru	Tanpa relasi	✓
		Menggunakan operator AND(*)	✓
		Menggunakan operator OR(/)	✓
		Menggunakan kombinasi AND(*) dan OR(/)	✓
2.2	Sistem dapat mengelola persetujuan ekuivalensi pada tiap-tiap mahasiswa	-	✓
2.2.1	Sistem dapat mengelola matakuliah yang wajib diambil pada kurikulum baru dan dapat menghapus krs lulus yang sudah pernah ditempuh	-	✓
2.2.2	Sistem menyediakan laporan matakuliah yang wajib diambil mahasiswa alihjenjang setelah melakukan simpan permanen	-	✓
2.2.3	Sistem dapat membatalkan simpan permanen sehingga data matakuliah yang wajib diambil masih bisa diubah	-	✓

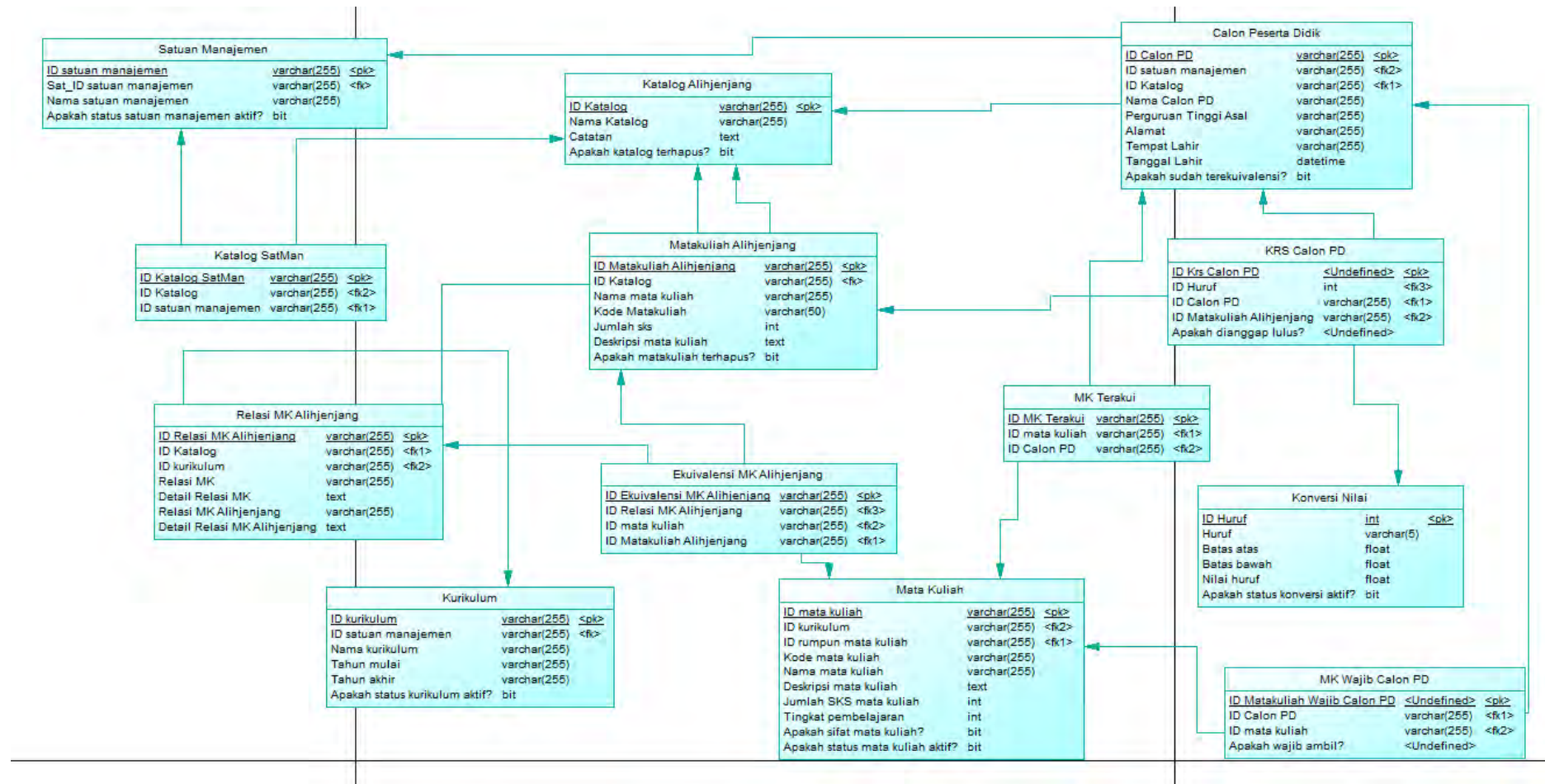
MAHASISWA

No	Kebutuhan Fungsional	Kondisi	Terpenuhi
1	Membuka laporan ekuivalensi yang sudah dilakukan mahasiswa	-	✓

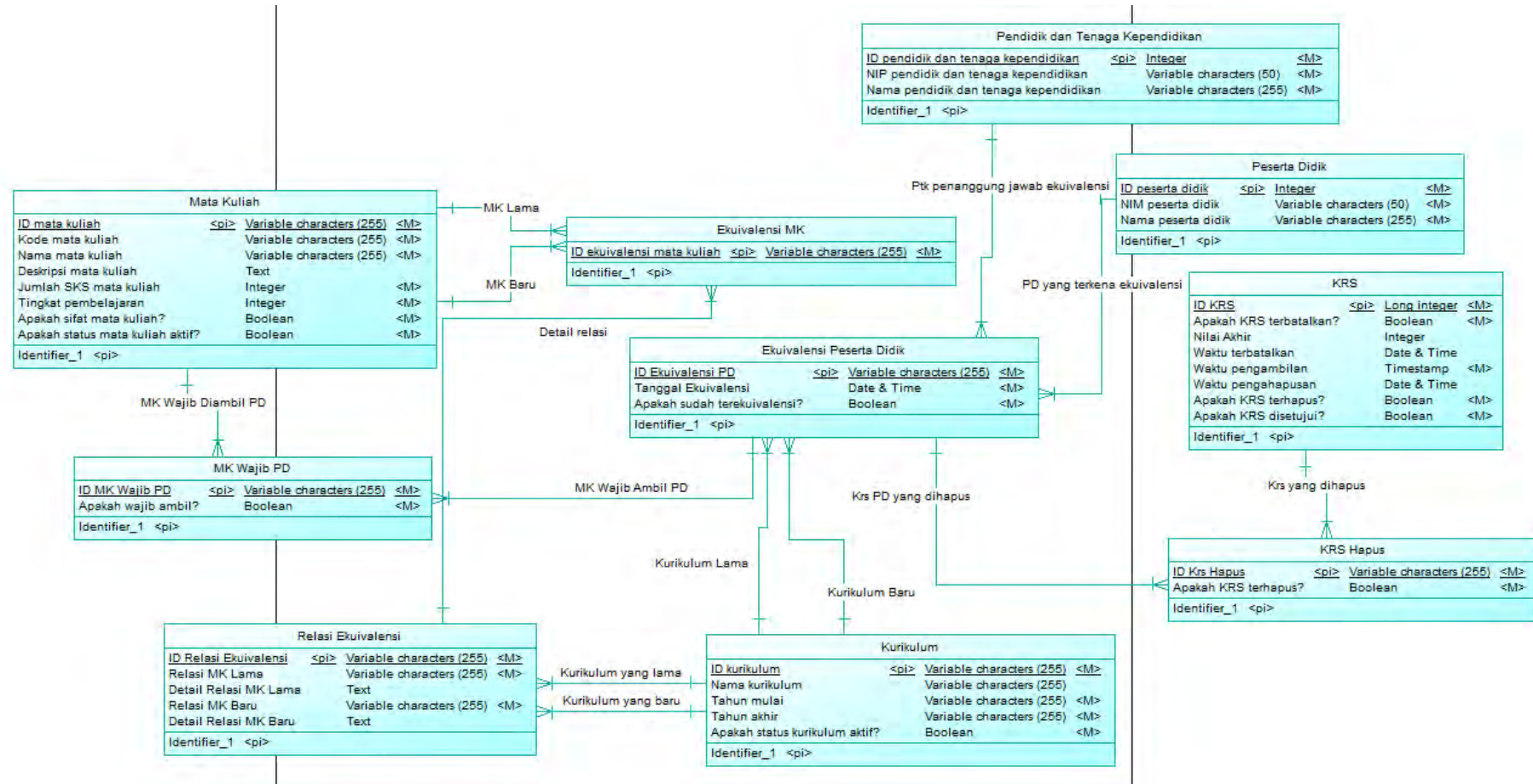
Gambar C.18 Lembar kuisioner



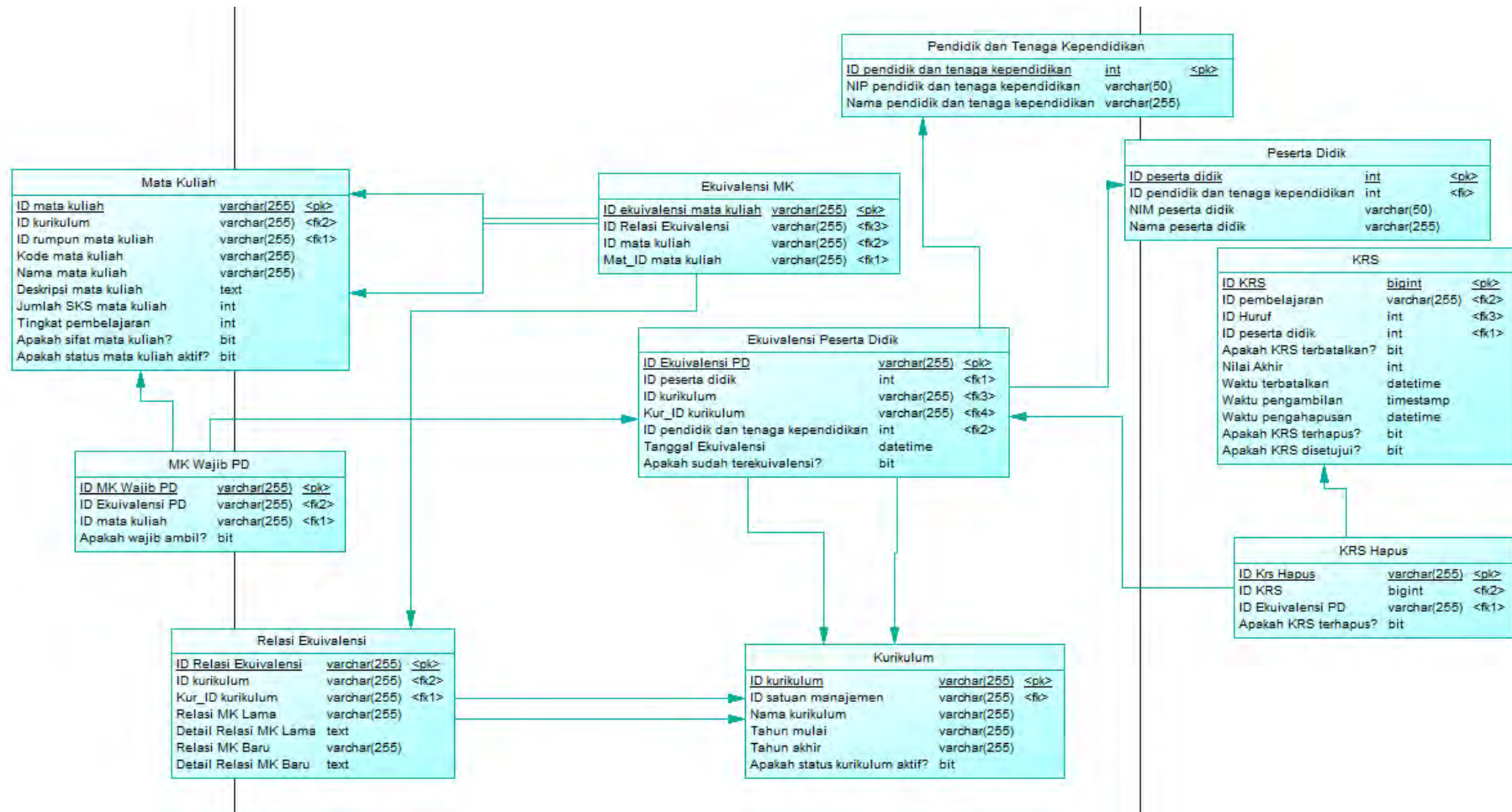
Gambar C.19 CDM ekuivalensi alih jenjang



Gambar C.20 PDM ekuivalensi alih jenjang



Gambar C.21 CDM ekuivalensi kurikulum



Gambar C.22 PDM ekuivalensi kurikulum

BIODATA PENULIS



Penulis, Bustan Amal Alfirdaus lahir di Kota Pati pada tanggal 29 Agustus 1993 dan dibesarkan di Kota Pati. Penulis adalah anak terakhir dari dua bersaudara.

Penulis telah menempuh pendidikan formal pada jenjang TK sampai dengan S-1 di TK Pertiwi Siwi Peni kab. Pati (1997-1999), SDN Winong 1 kab. Pati (1999-2005), SMPN 1 Winong kab. Pati (2005-2008), SMAN 1 Pati kab. Pati (2008-2011), Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya (2011-2015).

Di Jurusan Teknik Informatika, penulis mengambil bidang minat Rekayasa Perangkat Lunak (RPL) dan memiliki ketertarikan pada bidang *software development*. Penulis juga aktif dalam organisasi kemahasiswaan mulai dari HMTC (Himpunan Mahasiswa Teknik Computer-Informatika) sebagai Staf Departemen Pengembangan Sumber Daya Mahasiswa (2012-2013), Staf Ahli (2013-2014). Penulis pernah menjadi anggota Sponsorship Schematics 2012 dan menjadi Koordinator Perlengkapan dan Transportasi Schematics 2013. Penulis juga pernah menjadi asisten praktikum pada mata kuliah Pemrograman Terstruktur. Penulis dapat dihubungi pada alamat email bustan.amal.alfirdaus@gmail.com.